Descriptive Complexity Approaches To Inductive Inference

MS-CIS-91-08
GRASP LAB 253

Kevin Atteson

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

January 1991

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-98-

0745

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REP
	January, 1991	Final

4. TITLE AND SUBTITLE
Descriptive Complexity Approaches to Inductive Inference

5. FUNDING NUMBERS

6. AUTHORS
Kevin Atteson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Pennsylvania

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NI
4040 Fairfax Dr, Suite 500
Arlington, VA 22203-1613

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
See Attachment

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Descriptive Complexity Approaches to Inductive Inference

A Special Area Examination
by Kevin Atteson *
advised by Dr. Max Mintz

Abstract

We present a critical review of descriptive complexity approaches to inductive inference. Inductive inference is defined as any process by which a model of the world is formed from observations. The descriptive complexity approach is a formalization of Occam's razor: choose the simplest model consistent with the data. Descriptive complexity as defined by Kolmogorov, Chaitin and Solomonoff is presented as a generalization of Shannon's entropy. We discuss its relationship with randomness and present examples. However, a major result of the theory is negative: descriptive complexity is uncomputable.

Rissanen's minimum description length (MDL) principle is presented as a restricted form of the descriptive complexity which avoids the uncomputability problem. We demonstrate the effectiveness of MDL through its application to AR processes. Lastly, we present and discuss LeClerc's application of MDL to the problem of image segmentation.

*Acknowledgement: Navy Contract N0014-88-K-0630; AFOSR Grants 88-0244, 88-0296; Army/DAAL 03-89-C-0031PRI; NSF Grants CISE/CDA 88-22719, IRI 89-06770; the Dupont Corporation and Air Force LGFP SSAN 167 46 7002

Contents

1. Introduction

Human reasoning is commonly decomposed into two major categories. On the one hand, analysis or deductive inference can be defined as any process by which a model of the world is evaluated and its implications are made known. On the other hand, synthesis or inductive inference can be defined as any process by which a model of the world is determined from observations. Mathematics has had some success in the formalization of the deductive inference process through the framework of mathematical logic. This is not to say that all problems in deductive inference have been solved, but there is a formalization which can adequately state many problems of deductive inference. However, attempts to formalize the inductive inference process have been less successful.

In the mid 1960's, Kolmogorov[9], Chaitin[4] and Solomonoff[16] independently developed a formalization of inductive inference which is extremely general, that of descriptive complexity (elsewhere known as algorithmic complexity or Kolmogorov complexity; the term descriptive complexity is used here to avoid possible confusion with computational complexity). Unfortunately, in analogy with mathematical logic as a formalization of deductive inference, the formalization has little computational feasibility. In response to this problem, Rissanen[2] has formulated the minimum description length (MDL) approach to statistical inference (inductive inference with probabilistic models) which is basically a strengthening or a restriction in generality of descriptive complexity. Finally, we look at LeClerc's[3] application of the Rissanen formalism to image segmentation.

In the next 3 subsections, we present introductory material to help motivate each of the 3 major sections of the paper.

1.1. The Interpretation(s) of Probability Theory

1.1.1. Measure Theory

Measure theory, the abstract theory of volume, is now widely accepted as the axiomatic basis of probability theory. From the axioms of measure theory, various laws of probability can be derived. However, the axiom system does not directly answer the question, "What does it mean for some event to be random with a certain probability?" Several related questions are extremely important for inductive inference using probability models. How can we determine a probability for some event? How can we tell whether a particular set of data adequately characterized by a particular probability model? Section 2 of this paper examines attempts to define probability theory by answering these questions. There are several popular theories giving interpretations of probabilities. Such theories can be generally categorized as frequency-based approaches and subjective approaches. In subjective approaches, a probability is interpreted as a subjective degree of belief in some event. However, such theories are difficult to formalize and will not be presented here.

1.1.2. Probabilities as Frequencies

It has been long realized that probabilities often arise in connection with the frequency of occurrence of some event. Let us consider attempting to define probability in terms of frequencies of an event. In tossing a fair coin many times, one believes that the relative frequency of heads occurring will be approach $\frac{1}{2}$ with probability one. However, what is meant by with probability one? It seems that probability cannot be defined in terms of frequencies without creating a circular definition. Furthermore, frequency does not adequately characterize randomness. The frequencies of 0's and 1's in the binary representation of π converge to $\frac{1}{2}$ but π is not random.

1.2. Statistical Modeling and Inference

In section 3, we will be concerned with statistical inference. Statistical inference is essentially inductive inference restricted to probabilistic models. Some introductory material is presented here.

1.2.1. Model Classes and Nested Model Classes

In statistical modeling, we are interested in choosing, from some class of models, a probabilistic model which adequately characterizes the data. Broadly, there are two classes of statistical models, parametric and non-parametric. In this paper, we will be concerned only with parametric model classes. In parametric statistical inference, the class of models from which the choice is made are characterized by some parameter vector $\theta=(\theta_1,\theta_2,...,\theta_k)$ which ranges in a subset Θ^k of Euclidean k-space. For the statistical models in section 3, the data space will be time series, that is, finite sequences of real numbers. Such a sequence of points will be denoted by $x^n=(x_1,x_2,...,x_n)$ where n is the number of points and x^0 is the sequence containing no points. The *ith* position of the sequence will be called time step i. x_i is the value of the sequence at time step i. We will denote the distribution of x^n, given a model with parameter θ, by $f_\theta(x^n)$. No distinction will be made in the sequel between the specific model and the parameters which define it.

The parameters of a model class may come from a more general set than a subset of Euclidean k-space. Here, some general classes that we consider have a varying number of parameters, that is, each model is parameterized by a finite number of parameters but different models may have a different number of parameters. In other words, the parameters range in a set A defined as the union of subsets of Euclidean k-space over all k:

$$A = \bigcup_{k=1}^{\infty} \Theta^k$$

where Θ^k is the subset of Euclidean k-space over which the set of parameters with dimension k ranges. Thus, the set A is the set of all vectors of finite dimension. Another way of viewing this is that the number of parameters is itself a parameter. The model class will be called nested if the models with parameters of dimension k is a subset of the models with parameters of dimension k+1, that is:

$$\{f_\theta(x):\theta \in \Theta^k\} \subset \{f_\theta(x):\theta \in \Theta^{k+1}\}$$

In other words, the model classes are increase in generality as the dimension of their parameter vector increases. For an example of a nested model class, see the discussion of ARMA models below.

1.2.2. Statistical Inference Procedures

There are many formalisms by which a parametric model may be chosen from within some model class. In section 3, we present the minimum description length (MDL) approach to statistical inference. Here, we will present maximum likelihood estimation, the least squares criterion and maximum a posteriori probability estimation. In maximum likelihood estimation, we choose the parameter value, i.e., model, which is the most likely to have produced the data, that is, we choose θ to maximize the so-called likelihood function, $f_\theta(x^n)$. For example, let x^n be a sequence of independent, identically distributed Gaussian random variables with unknown mean μ and unit variance:

$$f_\mu(x^n) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu)^2}{2}\right)$$

In order to choose μ, we maximize the above with respect to μ which is equivalent to discarding the positive constant and maximizing the logarithm:

$$\ln \prod_{i=1}^{n} \exp\left(-\frac{(x_i - \mu)^2}{2}\right) = \sum_{i=1}^{n} \ln \exp\left(-\frac{(x_i - \mu)^2}{2}\right) = -\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2}$$

which is equivalent to choosing μ to minimize the sum of squared errors:

$$\sum_{i=1}^{n} (x_i - \mu)^2$$

This is the least squared error criterion. We choose the model such that predictions made by using the model minimize the sum of squared errors. As we have just seen, for Gaussian random variables with known variance, this is equivalent to maximum likelihood estimation (in fact, the solution of either optimization problem is the sample mean).

Thus, maximum likelihood estimation and the least squares criterion are two methods for choosing models. Unfortunately, these methods, as well as many of the other standard statistical methods, do not help in choosing from more general classes of models such as nested models. To see this, consider the model which gives probability one to the data. Under both methods discussed above, this model is the best possible model and will always be chosen if it is in the considered class. However, this model is not very satisfying and will generally have little predictive value. In the case of nested models, all models which may be parameterized by a k-vector are also parameterized by a (k+1)-vector, and so the likelihood of the best (k+1)-vector will be at least the likelihood of the best k-vector. Typically, the likelihood of the best (k+1)-vector will be greater than the likelihood of the best k-vector since there is more generality in the models of dimension k+1. Thus, maximum likelihood will typically choose arbitrarily high dimensional models.

On the other hand, the Bayesian formalism for statistical inference requires specification of a prior distribution on the parameters of the model. This prior distribution may represent prior or subjective knowledge as to which models are more likely to occur. We denote the prior distribution on the parameters, θ, by $g(\theta)$. Based on the prior distribution, we can compute the conditional distribution of the model given the data (here we denote the distribution of the data given the model by $f(x^n|\theta)$ as opposed to the previous $f_\theta(x^n)$):

$$g(\theta \mid x^n) = \frac{f(x^n \mid \theta)\, g(\theta)}{f(x^n)}$$

where $f(x^n)$ is the marginal distribution of the data with respect to the parameters:

$$f(x^n) = \int f(x^n \mid \theta)\, g(\theta)\, d\theta$$

(integrating over all of θ-space). In the maximum a posteriori method of estimation, we choose the model which maximizes the probability of the model given the data, that is, which maximizes $f(\theta|x^n)$. Note that this differs from the maximum likelihood method in which we choose the model in which the data would have the maximum probability of occurrence and no prior was required. In the equation for $f(\theta|x^n)$, we see that $f(x)$ in the denominator is constant with respect to θ (since it is just the normalizing constant) and so maximizing $f(\theta|x^n)$ is equivalent to maximizing:

$$f(x^n \mid \theta)\, f(\theta)$$

Finally, note that if the prior has the same value on all models θ (which is sometimes considered to be the case that there is no prior knowledge since it can be shown to be the distribution representing the least amount of prior "information"), this is equivalent to the maximum likelihood approach.

1.2.3. Example Model Class: ARMA Models

Autoregressive moving average (ARMA) models are probabilistic models which can be used for the analysis of time series, which, as mentioned, are finite sequences of real numbers. An ARMA process is a sequence of random variables which are derived from a white noise excitation source by stable linear filtering. White noise is a sequence of independent and identically distributed random variables with bounded second moment. An autoregressive (AR) process is obtained from white noise by passing it through an infinite impulse response (IIR) filter as demonstrated below:

$$x_n = \sum_{i=1}^{p} a_i \, x_{n-i} + e_n$$

where $\langle x_n \rangle$ is the AR process, $(a_1, ..., a_p)$ are the p coefficients of the IIR filter and $\langle e_n \rangle$ is the white noise process. The filter must be stable so that the process is stationary. An IIR filter is stable if its output sequence is bounded whenever its input sequence is bounded. A moving average (MA) process is obtained from white noise by passing it through a finite impulse response (FIR) filter as demonstrated below:

$$x_n = \sum_{i=1}^{q} b_i \, e_{n-i} + e_n$$

where $\langle x_n \rangle$ is the MA process, $(b_1, ..., b_n)$ are the q coefficients of the FIR filter and $\langle e_n \rangle$ is the white noise process. In order to maintain a dual relationship between AR and MA processes, it is required that the FIR filter of an MA process have a stable inverse filter. Thus, an ARMA process can be described as:

$$x_n = \sum_{i=1}^{p} a_i \, x_{n-i} + \sum_{i=1}^{q} b_i \, e_{n-i} + e_n$$

There is a large body of work on identification of ARMA processes[3]. However, most of the methods of identification, as discussed in the previous subsection, do not include a completely objective method for determination of the number of parameters (coefficients of the AR and MA filters). ARMA processes (more specifically AR processes) are used as an example in the discussion of the minimum description length principle since MDL is well-suited for determination of ARMA models, including the number of parameters.

1.3. Image Segmentation

In computer vision, we attempt to recognize objects in an intensity image (for our purposes, we will consider intensity images, although other types of images are sometimes available). The problem of image segmentation is often considered to be one of the stages in the process of vision. In image segmentation, we attempt to partition the image into roughly homogeneous regions, where homogeneous

can mean constant intensity, smoothly varying intensity, uniform texture, etc. After segmentation, we might attempt to identify each region as an object or sub-object of some type.

There are many approaches to the segmentation problem[6]. Some techniques are based on local information such as region growing, in which seed regions are expanded based on the similarity of neighboring pixels. Others techniques are based on more global information such as regularization[11], in which a segmented model of the image is chosen which minimizes a functional containing terms for similarity to the original image, smoothness of the model between segment boundaries, and the lengths of the boundaries. An approach to image segmentation based on the MDL principle is presented in section 4. However, in image segmentation in general, there is no criterion for the comparison of segmentations besides visual inspection with the exception of what will be described in section 4 of this report.

2. Complexity and Randomness

In this section, we discuss the theoretical framework for randomness given by descriptive complexity and related topics. We consider the example of sequences of i.i.d. symmetric Bernoulli random variables as the probability model. Such a sequence consists of 0's and 1's determined independently and occuring with equal probability, i.e. tosses of a fair coin.

2.1. The Partial Recursive Functions

In this report, the partial recursive functions are used as the model of computation. These are essentially the functions which can be implemented by a computer program (assuming an arbitrarily large amount of memory is available). The functions may be partial because for some values of inputs, a computer program may never halt to return an answer. The partial recursive functions are a countable subset of the set of all partial functions from the set of natural numbers into itself. The set of natural numbers is isomorphic to the set of strings over the alphabet {0, 1} (or any finite set) and the two sets will be used interchangeably. The concatenation of strings v and w, written vw, is the string containing the symbols of v followed by the symbols of w. For example, 01101 is the concatenation of 01 and 101. A string v is a prefix of a string w if there is a string u such that w=vu. For details on the construction of the partial recursive functions see the paper by Zvonkin and Levin[19], or any text in the theory of computation.

A partial recursive function which is total, i.e., one which is defined everywhere, is called general recursive. The length function is a general recursive function which will be important in the sequel. The length function will be denoted by L(x) and represents the length of the string x. For example, L(00)=2 and L(0110011)=7.

In much of what follows we will be interested in two-place partial recursive functions. There is a universal two-place partial recursive function, F, that is, a function F such that for any two-place partial recursive function, f(x,y), there is a string α_f such that:

$$F(\alpha_f x, y) = f(x, y)$$

for all strings x. In other words, this universal partial recursive function can "simulate" any other partial recursive function using the "program" α_f by prepending the program onto the first argument. A universal partial recursive function is roughly equivalent to a programming environment and is called a universal interpreter.

The partial recursive functions (and therefore the universal partial recursive function) are equivalent in computational power to Turing machines which are used elsewhere in the literature. The Church/Turing thesis is a conjecture which claims that either of these, as well as many other models of computation that have been found to be equivalent, can in fact compute anything that can be computed in principle. Thus, any algorithm which performs a finite string of operations on finite (but arbitrarily large) data, can in theory be described by a partial recursive function. Some of the algorithms in the sequel will be described in English and the corresponding partial recursive function will not be stated explicitly though it will always exist.

2.2. Preliminary Work

2.2.1. Von Mises' Collectives

In 1919, Von Mises[18] introduced the notion of a collective or random sequence. Von Mises chose a specific set of infinite sequences of 0's and 1's as the set of random sequences, i.e. sequences which are believable as samples from a symmetric Bernoulli random variable. The Von Mises definition is based on the idea that no gambling system should be able to turn the odds in favor of a gambler predicting the outcome of a fair coin toss. Thus, the frequency of 1's in the sequence should converge to $\frac{1}{2}$. Furthermore, no gambling system should be able to change this asymptotic frequency of 1's. In other words, any infinite subsequence chosen based a selection rule should have the same asymptotic frequency. The selection rules must have the property of being "proper", that is, each element of the subsequence is selected based solely on knowledge of the elements of the sequence prior to that element, as would be the case in any gambling system. In the first formulation, Von Mises merely suggests using a countable set of "proper" selection rules. However, the asymptotic frequency should be invariant to any proper selection rule which any gambler could compute. For this reason, in 1940, Church[5] suggested specifically using the set of all general recursive functions as the selection rules.

Von Mises definition of randomness suggests a connection between randomness and inductive inference. A string is random if any gambling system cannot help the gambler in predicting the future of the string. In other words, a string is random if it cannot be induced from its initial fragments. This suggests that a formal definition of randomness will inherently specify a formal definition of inductive inference and vice versa.

2.2.2. Kolmogorov's "Foundations of Probability Theory"

In 1933, Kolmogorov published his landmark book, "The Foundations of Probability Theory"[8]. This book was the first to present measure theory as the basis of probability theory. It had an overwhelming impact on probability theory. For our purposes, the book had the impact of turning attention away from the work of Von Mises. In fact, Kolmogorov himself did not accept Von Mises work. However, measure theory does not answer some fundamental questions about probability which can be traced back to Laplace. For example, suppose someone gives you the following two sequences and says that one of the sequences was produced by coin tossing:

1111111111 0011101100

Most people would choose the latter sequence as being more random. However, both sequences have exactly the same probability of occuring. It is ironic that it was Kolmogorov who, a few years after the death of Von Mises, finally presented a consistent theory of randomness and brought new life into the program of Von Mises.

2.2.3. Shannon's Entropy

In 1948, C. E. Shannon published a seminal work entitled "A Mathematical Theory of Communication". This work laid the foundation for what was to become modern information theory. It also had a large impact on the development of descriptive complexity theory. Shannon's entropy (which is itself a generalization of the concept from statistical mechanics) is essentially a restricted form of descriptive complexity

In his paper, Shannon considered communication of a finite set of symbols over a communications channel (Fig. 1). The theory is developed by providing a mathematical model of a communication system. The communication system consists of an information source, a communications channel, and a destination for the information. The information source which produces the symbols is assumed to be an ergodic Markov chain. The symbols which are produced are encoded before being transmitted over the channel. In our case, it is instructive to consider the channel to be a computer memory (which is merely a time delay channel). Shannon's theory allows one to determine how to encode symbols which are to be transmitted so that to minimize use of the channel. After transmission, the symbols are decoded and forwarded to the destination. A fundamental assumption of Shannon's work is that the distribution (the transition probabilities, etc.) of the source is known in full to the designer of the encoder and decoder. Shannon shows how the ergodic Markov chain can be unfolded into a independent and identically distributed random sequence of "long" strings. Thus, we can assume that the source is a random variable, X, with range $x_1,...,x_n$ having probabilities $p_1=Pr[X=x_1],...,p_n=Pr[X=x_n]$.

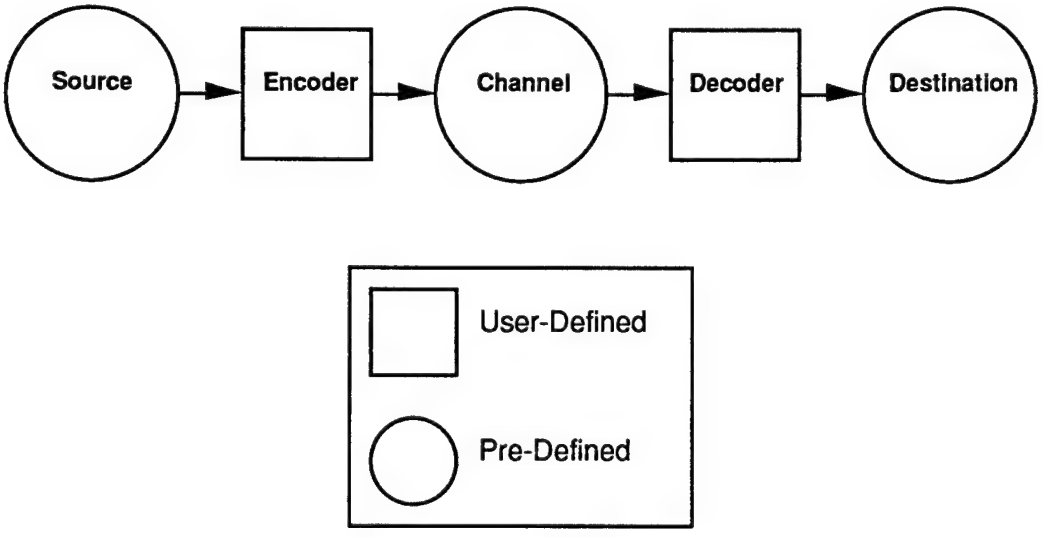
We wish to design an encoder which assigns a string of symbols as the code for each symbol produced by the source. It is desirable that the code length be as short as possible in order to minimize use of the channel which is assumed to be the limiting resource. A prefix code is a code for which no code string is a prefix of another code string. This condition is required if the code is to be decodable. Let $L(x_i)$ be the length of any code of the source symbols. Shannon's entropy, $H(p_1,...,p_n)$, is a lower bound on the expected number bits per symbol of any prefix code of the source symbols:

$$E[L(X)] \geq E[-\log_b(Pr[X])] = H(p_1, \ldots, p_n) = -\sum_{i=1}^{n} p_i \log_b(p_i)$$

where b is the number of symbols available to the decoder (here, we use binary codes and so b=2). Thus, in order to create a code with small expected code length, it is desirable to choose the code length for each x_i to be as close as possible to $-\log_2(p_i)$. Shannon also shows how this lower bound can be achieved to within an arbitrarily small precision by encoding long strings of source symbols. Note that another part of Shannon's information theory yields an upper bound for the expected number of "bits of information" which can be transmitted over a given channel per unit time but this part of information theory is not relevant to the current report.

Figure 1

Shannon's Communication System



One important aspect of Shannon's work is that the entropy represents the size of the minimum length encoding for strings given a certain probability distribution. Thus, the entropy is a measure of how much regularity is contained in strings from a given distribution. Distributions with little regularity have high entropy and are "uncertain" in that it is difficult to accurately guess which symbol will be produced next. In fact, this is the central idea behind the maximum entropy principle. Sometimes it is desirable to derive

a probability distribution which expresses ignorance about some events (for example, to derive a prior when there is no known prior). The maximum entropy principle asserts that the way to express this ignorance is to choose that probability distribution which has the maximum entropy. This distribution is the one which is the most "uncertain" or "random".

2.3. Descriptive Complexity

2.3.1. Definitions and Invariance

Kolmogorov was interested in the generalization of the concept of entropy to more general sources. In 1965, he arrived at the extremely general notion of descriptive complexity. The descriptive complexity of a string x with respect to a partial recursive function f is defined as follows:

$$K_f(x) = \min_{\{p:f(p)=x\}} L(p)$$

that is, the length of a smallest program (input to the function) on which f outputs the string x. The descriptive complexity of a string x given a string y with respect to f is defined as follows:

$$K_f(x \mid y) = \min_{\{p:f(p,y)=x\}} L(p)$$

that is, the length of a smallest program (first argument to the function) which outputs the string x under f when given y as input (second argument to the function). Clearly, $K_f(x)=K_f(x|\varepsilon)$ where ε is the empty string (actually, any constant string would work). Thus, we develop some of the theory for $K_f(x|y)$ with the implication that the corresponding results hold for $K_f(x)$.

Note that the descriptive complexity is dependent upon the function f. However, for any universal partial recursive function, F, the following inequality applies:

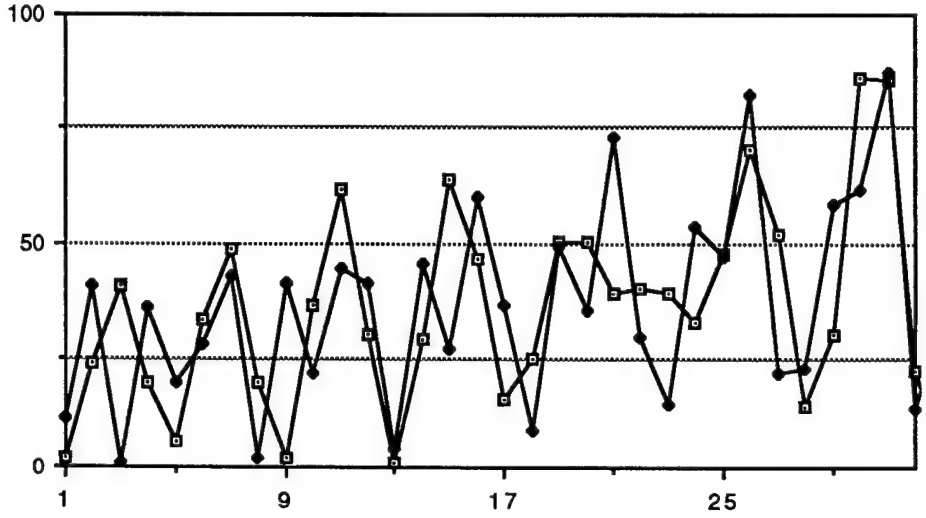
$$K_F(x \mid y) \leq K_f(x \mid y) + L(\alpha_f)$$

since in the worst case, f(x,y) can be simulated by $F(\alpha_f x, y)$. The important point here is that $L(\alpha_f)$ is independent of x and so the descriptive complexity with respect to F grows at most as fast as that of f. By the same token, the descriptive complexity with respect to any two universal partial recursive functions will grow at the same rate. Thus, in this sense, the descriptive complexity is invariant of the underlying interpreter. This fact is called the invariance theorem. Note also that growing at the same rate in this manner is in some sense stronger than the more commonly used rates of growth denoted by o, O and Θ since this is with respect to an additive constant . Figure 2 demonstrates some differences between the two notions of rate of growth. Θ-growth is strictly asymptotic and requires no correspondence between the initial segments of the functions. Also, Θ-growth only requires that a constant multiple of

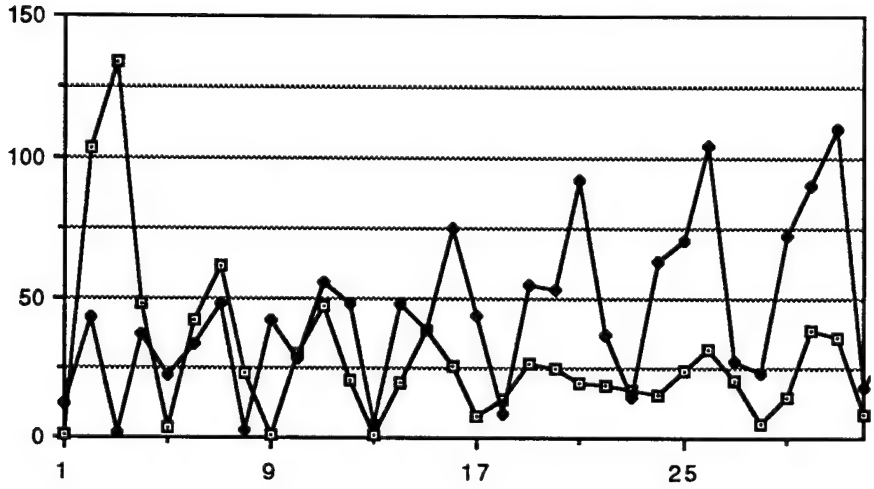
one of the functions approaches the other asymptotically whereas for additive growth, the constant is 1 (assuming that the functions tend to infinity which is the case here).

Figure 2

Descriptive Complexity Invariance
(functions equivalent in the same sense as descriptive complexity with respect to two different universal interpreters are equivalent)



Θ Invariance
(functions which are Θ equivalent)



It is easy to see that descriptive complexity is a generalization of entropy to a more general set of information sources, i.e. the partial recursive functions. Whereas entropy is the minimum average encoding length when the probability distribution is known to the sender and receiver, descriptive

complexity is the minimum encoding length when a common universal interpreter is known to the sender and receiver. In fact, the Church/Turing thesis purports that the partial recursive functions are the most general form of information source. Thus, in this sense, descriptive complexity is the most general form of entropy or encoding length possible.

2.3.2. Randomness

One particularly important upper bound of descriptive complexity is the length function. Consider the projection function $\pi(x,y)=x$. π is partial recursive and so:

$$K(x \mid y) \leq K_{\pi}(x \mid y) + L(\alpha_{\pi}) = L(x) + L(\alpha_{\pi})$$

the descriptive complexity is bounded by the length function to within an additive constant. This is a very intuitive notion. The objects which are most difficult to describe cannot be described in any way other than listing them in full and so their minimum description length, i.e. their descriptive complexity, is roughly equal to their length (the constant is the length of the minimum length code to tell the interpreter that a literal value follows). Kolmogorov had more to say about these complex strings.

Consider the set of "long" strings with descriptive complexity near their lengths:

$$\{x: K_F(x) \geq L(x) - c\}$$

for some constant c. Kolmogorov suggested that these are the strings which should be considered random. For these strings, there is essentially no shorter way to describe them than by listing them. This is analogous with the maximum entropy principle in that randomness is the property of only being described by a large number of bits. Notice that this gives an implicit definition of inductive inference. In order to choose a model of a string, we choose the smallest program which can produce that string.

If such a string were to have any regularities, the regularities could be used to describe the strings with a shorter program. For example, suppose the frequency of 1's in a string x is greater than the frequency of 0's in x. Let p be the frequency of 1's. As Shannon proved, for any ε, if the strings are large enough, the strings of symbols (in this case {0, 1}) can be encoded so that the average number of bits to encode a symbol is:

$$H(p, 1 - p) + \varepsilon = -p \log_2 p - (1 - p) \log_2(1 - p) + \varepsilon$$

This function H(p, 1 - p) has a unique maximum of 1 at $p=\frac{1}{2}$. Thus, if $p>\frac{1}{2}$ then $H(p,1-p)<1$ and we can choose ε so that:

$$H(p, 1 - p) + \varepsilon < 1$$

The total length of encoding a sufficiently long string will therefore be:

$$L(x)\,(H(p, 1 - p) + \varepsilon)$$

Thus, one possible program to generate the string would list the encoding of the string as part of the program and would merely decode this to obtain the string. The total length of this program would be the length of the encoding plus the length of the program to do the decoding:

$$K_F(x) \leq L(x)\,(H(p, 1 - p) + \varepsilon) + L(\alpha)$$

where α is the program to decode the encoded string. Since the constant multiplier of L(x) is less than 1, this will be less than L(x)-c for large enough L(x), i.e. for a long enough string. Therefore, long enough strings with frequency $p \geq \frac{1}{2}$ would not be random in the sense of Kolmogorov. Similarly, if the frequency of any short substring (short relative to the length of the string), is more than $\frac{1}{2^k}$ where k is the length of the substring, then the same technique could be used to find a shorter description of the string.

As a further example, consider a string which is a picture with k-fold symmetry. This string also is not random. We can write a program which lists one of the symmetric pieces and reproduces it k times to produce the string. The length of this program would be:

$$K_F(x) \leq \frac{1}{k} L(x) + L(\alpha)$$

where α is the program to draw a picture with k-fold symmetry. Again, for large enough L(x), the above length would be less than L(x)-c (actually, this example could be subsumed by the previous one in certain circumstances). The thesis here is clear, any regularity can be captured by a program and therefore, the random strings in the sense of Kolmogorov are strings containing no regularities.

Now, let us attempt to count the number of random strings of length n. First note that there are exactly $2^{n-c}-1$ strings of length less than n-c. This is of course the number of programs of length less than n-c. Thus, there are <u>at most</u> $2^{n-c}-1$ strings with complexity less than n-c. So of the 2^n strings of length n, there are at most $2^{n-c}-1$ which are not random (in reality there may be many less than this since many of the programs may produce strings of different length or produce repeats). Therefore, most of the strings of length n are random. The same basic idea can be applied to infinite strings (i.e., real numbers) and the random strings can be shown to have Lebesgue measure 1 in [0,1] (of course, the theory has to been modified somewhat for infinite strings).

2.3.3. Computing the Descriptive Complexity

The descriptive complexity function is not partial recursive. To prove this by contradiction, assume that the descriptive complexity is partial recursive. First note that the descriptive complexity with respect to a function with infinite codomain is unbounded. This is because, for any given length, the number of programs of that length is finite, and so there is only a finite number of strings with at most that complexity. Thus, an infinite set of strings will have strings of arbitrarily large complexity. Next note that the following inequality holds:

$$K_f(f(x, y) \mid y) \leq L(x)$$

since x is a program of length L(x) generating f(x,y) under f with input y. Now consider some universal interpreter F. Define f(x,y) so that $x \leq K_F(f(x,y)|y)$ as follows:

$$f(x, y) = \min_{\{z:K_F(z|y)\geq x\}} z$$

Now if K_F is partial recursive then so is f since $\geq$ can be defined as a partial recursive function (relation) and minimization of partial recursive functions are partial recursive. Therefore, we have:

$$x \leq K_F(f(x, y) \mid y) \leq K_f(f(x, y) \mid y) + L(\alpha_f) \leq L(x) + L(\alpha_f)$$

But this is a contradiction for large enough x since L(x) grows at the same rate as $\log_2(x)$. Therefore, K_F cannot be partial recursive. In the remainder of this report, proofs will be omitted for brevity.

Despite this condition, descriptive complexity can be approximated. For example, it was previously shown that the length of a string (with an additive constant) acts as an upper bound on its descriptive complexity. There are in fact many other ways in which the descriptive complexity can be approximated. Any type of regularity which can be computed is an upper bound to the descriptive complexity such as the entropy and symmetry tests described above.

Let us approximate the descriptive complexity of some string x by finding programs which describe x. Again fix a universal interpreter F. The function $L(x)+\alpha_1$ provides an upper bound on the size of a minimal length program producing x. Suppose we take each possible program of size less than this and allow each to run for some predetermined number of time steps t. Let H(x,y,t) be the length of the first program which describes x on input y and finishes in t time steps (time step will not be defined rigorously here but any notion of time is sufficient). If none of the programs which finishes in time t produce x then we let $H(x,y,t)=L(x)+\alpha_1$. Note that H(x,y,t) is general recursive. Since H(x,y,t) is the length of a program which describes x, it is an upper bound for K(x|y). For any t, there could be a program which is shorter than H(x,y,t) but describes x in more than t time steps. However, for t larger than the length of time it takes a shortest program to describe x on input y, H(x,y,t)=K(x|y). Thus, we have the following:

$$K(x \mid y) \leq H(x, y, t)$$

$$\lim_{t \to \infty} H(x, y, t) = K(x \mid y)$$

and so H(x,y,t) approximates K(x|y) from above. It is important to note that there is no general recursive function which approximates the descriptive complexity from below in the same manner that H(x,y,t) approximates it from above. In fact, there are no general recursive functions which bound the descriptive complexity from below and tend to infinity.

2.4. Martin-Löf's Tests and Randomness

In 1966, Martin-Löf introduced a definition of randomness based on an abstraction of statistical testing. This definition was constructed so as to be applicable to finite sequences. A statistical test is a rule for determining whether a set of data is fitted by some probabilistic model. The test may either accept or reject a given set of data. The level of significance of such a test is the probability that data which is generated by the model is rejected by the test (probability of false rejection). Martin-Löf develops the theory for tests which determine whether data is fitted by symmetric Bernoulli distribution and then generalizes it to arbitrary probability spaces. Here we will concentrate on the theory of randomness for Bernoulli distributed random variables. The presentation here has been modified from those found in the literature to fit the scope of this report.

A test of regularity for strings is a partial recursive function, g(x,m), which terminates with g(x,m)=0 when its input x should be accepted as having regularity g at the level 2^{-m} and is undefined otherwise. In symmetric Bernoulli trials, all strings of the same length are equiprobable. Since 2^{-m} represents the level of the test, the following must hold:

$$\mid \{x: L(x) = n \text{ and } g(x, m) = 0\} \mid \leq 2^{n-m}$$

that is, a fraction 2^{-m} of the strings of length n are accepted. Also, if a string is accepted at level 2^{-m} then it should also be accepted at level 2^{-p} for all $p \leq m$ and so:

$$p \leq m \Rightarrow \{x: g(x, m) = 0\} \subseteq \{x: g(x, p) = 0\}$$

A test of regularity is any such function in which the above two properties hold. For example, the function might test for the proximity of the frequency of 1's to $\frac{1}{2}$ or it might test for certain values of the average length of runs of 0's or 1's or anything which can be computed with partial recursive functions. For any given test of regularity, the largest level at which a string x is accepted is known as the randomness defect, $m_g(x)$, of the string:

$$m_g(x) = \max_{\{m:g(x,m)=0\}} m$$

The randomness defect will be one less than the first level at which the string x is rejected. By the first property of tests, there can be at most $2^{n-(n+1)}=\frac{1}{2}$ strings of length n and which are rejected by a test at level n+1 and so there are no strings accepted at this level. Therefore, $m_g(x) \leq n$, that is, the length function bounds the randomness defect for any test g (no additive constant here).

Analogous to the universal interpreter for descriptive complexity, there is a universal test of randomness, G. The test is universal in that for any test g, the following holds:

$$m_g(x) \leq m_G(x) + c_g$$

In other words, if g rejects x at some level, then G rejects x at a level which is at most some constant number higher. A string is random in the sense of Martin-Löf if it has no regularities, that is, if it is rejected by every possible test at some small level. This definition would be unwieldy were it not for the existence of a universal test. From the above property, a string is rejected by the universal test at some small level if and only if it is rejected by every possible test at some higher level (which depends on the test). Thus, the randomness defect with respect to a universal test gives an indication of the non-randomness of a string. A string with small randomness defect is rejected at a small level and so does not contain any regularities with a reasonable certainty. This can be extended to give a precise definition of randomness for infinite strings. An infinite string for which the randomness defect of its initial segments is bounded is random.

Martin-Löf went further to show a relationship between the randomness defect of a string x with respect to a universal test G and the descriptive complexity of x given the length of x, with respect to a universal interpreter F:

$$| L(x) - m_G(x) - K_F(x \mid L(x)) | \leq c$$

for some constant c. Thus, the randomness defect and the descriptive complexity have an additive inverse relationship. Since L(x) bounds both $m_G(x)$ and $K_F(x|L(x))$ (approximately), when either one of the functions is near this maximal value, the other must be small. This relates our two notions of randomness. Kolmogorov defines random strings as strings which have complexity close to their length. Martin-Löf defines random strings as strings with small randomness defect. By the above inequality, we see that these two definitions correspond closely (the use of $K_F(x|L(x))$ may actually be more appropriate than $K_F(x)$ because randomness is relative to other strings of the same length). Random strings are strings which cannot be described concisely and these are precisely the strings which have no regularities. Just as belief in the Church/Turing thesis is strengthened by the correspondence between the two

seemingly different forms of computation, one's belief in these definitions of randomness is strengthened by the correspondence of the two definitions.

2.5. Discussion

Von Mises was the first to introduce an explicit definition of randomness in terms of a set of random sequences. Implicit in his formulation was the importance of the relationship between inductive inference and randomness. I believe that this is essentially the correct notion of randomness. Random strings are those strings upon which a gambler cannot bet and win consistently; those strings for which the future cannot be guessed from the initial segments. However, Von Mises original definition was somewhat informal and in fact, formalizations of it were found to be inconsistent[19].

Descriptive complexity theory goes even further in demonstrating the relationship between inductive inference and randomness. With descriptive complexity approach to inductive inference, we choose the model which allows the data to be communicated as concisely as possible. In this theory, randomness is defined in terms of the descriptive complexity which measures how simple of a model a string has, that is, how well it can be induced. If the string is random, its only models are essentially restatements of the string. The descriptive complexity approach to inductive inference is extremely general and, as is often the case, pays for its generality with practicability. The descriptive complexity is not computable. We can approximate it from above by various means but we cannot approximate it closely from below. This essentially means that we can determine when a string is non-random but we cannot determine when a string is random. In other words, we can find certain regularities in strings but we cannot determine when a string has no regularities.

Another problem with descriptive complexity is in choosing an interpreter. The invariance theorem implies that the complexity will not differ significantly between any *two* universal interpreters for sufficiently long strings (or more directly, sufficiently complex strings). This can be extended to any finite number of universal interpreters. However, when we consider the countable number of universal interpreters, we find that the strings must be arbitrarily long for the invariance theorem to hold. To clarify this, consider the case of a fixed finite piece of data. For any number, we can choose a universal interpreter in which this piece of data has that complexity. Thus, for the case of finite data, the choice of the interpreter can make the data random or non-random. The complexity is still invariant for the case of infinite data, or data which can be extended indefinitely. However, in practice we are often limited in the number of measurements we can take due to various constraints. Perhaps one solution is to choose an interpreter which is suitable for the problem at hand, i.e. one that codes the type of regularities which are expected with small codes. In fact, it may well be that randomness is context dependent. For instance, when one considers the set of all pictures, it may be that the picture encoded by the first n binary digits of the number π will be a random picture, but this same data will be non-random as a long binary string. However, this context dependency should be captured by the underlying distribution (the results above can be extended to probability spaces other than symmetric Bernoulli distributions). It is contrary to the

original intention of finding a general theory of randomness to use different interpreters in different contexts. Similarly, there are strings which are random for most realistic applications but which are non-random according to descriptive complexity theory. For example, the binary digits of π and the numbers produced with a random number generator demonstrate many features of randomness and yet are not random in that they have small programs which describe them.

There may also be other factors which should be included in a theory of complexity and randomness. For example, a certain string may be completely predictable but it might be computationally or practically infeasible to predict it. For example, if we were to precisely measure the initial velocities, mass distribution and surrounding air currents at the toss of a coin, we might be able to compute the face on which it lands, however, it is unlikely that we would be able to make the required measurements and computations to make this prediction before the event occurred (actually, this example is somewhat contrived since we would have to include the air currents, etc. in the program to compute the coin toss and so the coin toss would still be random). As another example, the 1,000,000th prime can be produced by a relatively short program but this program would be computationally expensive. It is unlikely that you would be able to produce the next digit of the 1,000,000th prime in time to win a bet. However, again, this is an external factor specific to the situation at hand. Thus, descriptive complexity defines a general approach to the problem of randomness but there may be other factors to consider in applications.

The definition of randomness via Martin-Löf's tests is essentially equivalent to that via descriptive complexity. As mentioned previously, this adds much weight to the descriptive complexity notion of randomness. However, since these definitions are equivalent, each of the points mentioned above apply equally to both. Martin-Löf's tests are not invariant to the choice of universal test for finite data and there may be factors not accounted for in specific applications.

3. Minimum Description Length Modeling

3.1. Introduction to MDL

The uncomputability of the descriptive complexity makes descriptive complexity theory impractical. The minimum description length (MDL) principle for statistical inference introduced by Rissanen[2] solves this problem of computability. MDL is concerned with detecting a certain class of regularities rather than the absence of all possible regularities which is impossible to detect as descriptive complexity theory proves. The type of regularities which are detected is restricted to certain probabilistic regularities generated by a chosen class of probabilistic models. This formalism can be seen as choosing a specific non-universal interpreter to determine the descriptive complexity. Although the interpreter is not as general as possible, the complexity of objects with respect to it is computable. Since the regularities detected with MDL are probabilistic, they do not give an exact description of the data. According to MDL, statistical inference is performed by choosing the model from a selected class which completely describes the data with the

smallest possible encoding. However, in order to completely describe the data within the chosen model class, it is often necessary to encode the model.

MDL requires selection of a class of probabilistic models. The specific class chosen depends on the specific problem but certain conditions must hold for the chosen class. The class of models must be indexed by some parameters. One of the strengths of Rissanen's formalism over other methods of statistical inference is that the number of parameters of the model may itself be a parameter, or, in other words, the class of models may have a varying number of parameters. Thus, MDL may be used with nested model classes. For example, the ARMA models would be one such class. ARMA models may have any number of AR parameters and any number of MA parameters and so the number of AR or MA parameters in a specific model can be viewed as a parameter. Recall that x^n refers to the n-vector of data $(x_1,...,x_n)$. In what follows, the distribution of a model from the model class will be denoted by $f_\alpha(x^n)$ where $\alpha=(k,\theta)$ represents the parameter vector θ as before and k the dimension of θ. Thus, the number of parameters is considered as a parameter, although it is treated differently in certain cases. Since we are concerned with coding the data, they must be finite sequences of elements from a countable set (we have only a countable number of codes). If the data elements are real numbers, Rissanen suggests discretizing the model class by choosing a precision r and integrating over regions of volume r^n to define a new model with a countable number of elements. For example, in the simple case in which each data element is a real number and x^n is distributed according to $f_\alpha(x^n)$, we define a new model with point mass function $\hat{f}_\alpha(\hat{x}^n)$ on the finite precision elements $\hat{x}^n=(\hat{x}_1,\hat{x}_2,\ldots,\hat{x}_n)$:

$$\hat{f}_\alpha(\hat{x}^n) = \int_{\hat{x}_n}^{\hat{x}_n+r} \cdots \int_{\hat{x}_1}^{\hat{x}_1+r} f_\alpha(x^n)\, dx_1 \cdots dx_n$$

In practice, this is not a problem since we never actually have data which are real numbers, that is, all data has some finite precision (otherwise, we would not be able to record it). In the sequel, I will use $f_\alpha(x^n)$ as the discretized version of the distribution rather than $\hat{f}_\alpha(\hat{x}^n)$ in order to spare notation.

3.1.1. The Non-predictive and Predictive Complexities

Given the class of models, Rissanen introduces non-predictive, semi-predictive (which is not described here) and predictive complexities. The three cases correspond to different methods of encoding the data and parameters. In the non-predictive method, the parameters are determined for the whole set of data. In Shannon's method of coding, the coding scheme is determined once based on the known distribution of the data before any data is sent and it is assumed that this scheme is built into both the encoder and decoder. The difference between Rissanen's non-predictive method of coding and Shannon's method is that with Rissanen's non-predictive method of coding, the coding scheme (or equivalently, the distribution of the data) is not assumed to be known by the decoder before transmission and so the parameters of the chosen model (which determine the coding scheme) are transmitted before the actual coded data. Thus, to completely specify the data, we must encode the model, i.e., the parameters, and

encode the data within the chosen model. If the model has distribution $f_\alpha(x^n)$ for some fixed parameters $\alpha=(k,\theta)$, then, according to Shannon, that model allows us to encode the data x^n in an asymptotic average length of $-\log_2 f_\alpha(x^n)$ bits. The total encoding length is therefore:

$$-\log_2 f_\alpha(x^n) + L(\alpha)$$

where $L(\alpha)$ is the length required to encode the parameters $\alpha=(k,\theta)$. For fixed k, θ must come from a countable set in order to encode it. Rissanen makes the assumption that θ is from a compact subset of Euclidean k-space and that the distribution $f_{(k,\theta)}(x^n)$ is smooth (twice continuously differentiable) with respect to θ. In order to make θ space countable, we choose a precision to which the parameters will be encoded. If the precision is too coarse, then we will not be able to accurately specify the model and the encoding length of the data will be large. On the other hand, if the precision is too fine, then the encoding length of the parameters will be long. Rissanen finds an optimal precision at which the parameters should be encoded for minimum description length (this is embodied in the theorem given in the next subsection). At this precision, the encoded parameters require:

$$L(\alpha) = \frac{k}{2}\log_2 n + o(\log_2 n)$$

bits. We can ignore the $o(\log_2 n)$ term since this is dominated by the $\log_2(n)$ term asymptotically. Thus, the total number of bits required to describe the data is the number of bits required to describe the model plus the number of bits required to describe the data in that model or:

$$L(\alpha, x^n) = -\log_2 f_\alpha(x^n) + \frac{k}{2}\log_2 n$$

This is the non-predictive complexity. We choose the model by minimizing the number of bits required to encode the data, that is, by choosing $\alpha=(k,\theta)$ which minimizes the above expression. In practice this involves minimizing $L(\alpha, x^n)$ over θ for each fixed k (within some reasonable range, for example, from 0 to n) and choosing the k for which $L(\alpha, x^n)$ is minimum with respect to θ. Note that for each fixed k, this corresponds exactly to the maximum likelihood method of parameter estimation.

Now we consider the predictive complexity. In the case of predictive complexity, a new model is determined for each point of data in a predictive manner. Note that this differs more significantly from Shannon's method of coding. In this case, the coding scheme is chosen adaptively, that is, a completely new coding scheme is chosen after each data point is transmitted. This corresponds to sending the data one at a time as opposed to sending them in a batch as in the non-predictive complexity. At each step, the model is chosen based only on preceding data points, so that the model can be determined by the decoder based on the data already sent. Thus, there is no need to transmit the model (the parameters).

Denote the model chosen at time t by $\hat{\alpha}(t)$. The probability that x_{t+1} will occur based on the chosen model is $f_{\hat{\alpha}(t)}(x_{t+1}|x^t)$. The length of encoding of x_{t+1} will therefore be $-\log_2 f_{\hat{\alpha}(t)}(x_{t+1}|x^t)$. The total encoding length will be the sum of the encoding lengths of the individual points:

$$-\sum_{t=0}^{n-1} \log_2 f_{\hat{\alpha}(t)}(x_{t+1} \mid x^t) = -\sum_{t=0}^{n-1} \log_2 f_{(\hat{k}(t);\, \hat{\theta}(t))}(x_{t+1} \mid x^t)$$

This is the predictive complexity. The procedure for choosing $\hat{\alpha}(t)$ for each t is derived from this using the constraint that only data from preceding time steps can be used in determining the model.

To summarize construction of the model in the predictive complexity case, the basic idea is to hold constant the object which is to be determined and minimize the resulting complexity. Thus, the determination of the model is performed in two steps (Fig. 3). In the first step, a value of k is held fixed over time and for each time step t, $\hat{\theta}_k(t)$ is chosen by holding θ fixed over time and minimizing the complexity. This is done for all reasonable values of k (say, between 0 and t-1). Then, the value of k is chosen such that when k is held fixed, the complexity with respect to the models $\hat{\theta}_k(t)$ is minimized. Notice that this method of model determination is well-suited for iterative procedure which predicts the data in real time (this is exactly because it is the predictive complexity). As a final consideration, the predictive complexity (or non-predictive complexity in the previous case) can be used to compare different classes of models.

Figure 3

Choosing a Model Using the Predictive Complexity

Step 1:

For each value of k from 1 to n:
For each value of t from 1 to n:
Choose $\hat{\theta}_k(t)$ to minimize:

$$\hat{\theta}_k(t) = \arg\min_{\theta} \left(-\sum_{i=0}^{t-1} \log_2 f_{(k;\theta)}(x_{i+1} \mid x^i) \right)$$

Step 2:

For each value of t from 1 to n:
Choose $\hat{k}(t)$ to minimize:

$$\hat{k}(t) = \arg\min_{k} \left(-\sum_{i=0}^{t-1} \log_2 f_{(k;\, \hat{\theta}_{k(i)})}(x_{i+1} \mid x^i) \right)$$

Predictive Complexity:

The predictive complexity is calculated as:

$$-\sum_{t=0}^{n-1} \log_2 f_{(\hat{k}(t);\, \hat{\theta}_{k(t)})}(x_{t+1} \mid x^t) = -\sum_{t=0}^{n-1} \log_2 f_{\hat{\alpha}(t)}(x_{t+1} \mid x^t)$$

In his later papers[14], Rissanen introduces the stochastic complexity which provides a unifying framework for the other types of complexity. The stochastic complexity equals $-\log_2 f(x)$, the size of encoding x, where f(x) is the marginal distribution of x derived from $f(x|\theta,k)$ by using priors $\pi(\theta|k)$ and Q(k) (he also discusses methods for choosing priors when they are not known in advance). He then suggests that the non-predictive, semi-predictive and predictive complexities are upper bounds approximating this stochastic complexity by implicitly choosing certain priors.

3.1.2. A Lower Bound on the Complexity

Rissanen gives a lower bound on the expected encoding length or complexity of data distributed according to $f_\alpha(x^n)$ with fixed $\alpha=(k,\theta)$ with θ from a compact subset of Euclidean k-space. He gives certain sufficient conditions on $f_\alpha(x^n)$ in order for the lower bound to hold. One condition is that $f_\alpha(x^n)$ is twice continuously differentiable with respect to θ. Also, a sequence of estimates of θ, $\hat{\theta}(x^n)$, must exist which satisfy the hypothesis of the central limit theorem, that is, there are constants $\delta(n)$ with:

$$\Pr\{ \sqrt{n} \, \| \hat{\theta}(X^n) - \theta \| \geq \log_2 n \} < \delta(n)$$

such that:

$$\sum_{i=0}^{\infty} \delta(n) < \infty$$

In other words, there must exist estimates such that their probability distributions do not have too much mass in the tails. Let $L(x^n)$ be the length of encoding the data x^n for any prefix code of the data x^n. The coding scheme is regular if $2^{-L(x^n)}$ satisfies the compatibility conditions for a stochastic process. The lower bound on the expected coding length of a regular code, $L(x^n)$, is given by:

$$E\left[L(X^n)\right] \geq E\left[-\log_2 f_\alpha(X^n)\right] + \left(\frac{1}{2} - \varepsilon\right) k \log_2 n + o(\log_2 n)$$

for any ε, for all but finitely many n and for all θ except in a set of Lebesgue measure 0.

The theorem is similar in nature to Shannon's theorem which states that the entropy of the source is the least possible expected code length per symbol. The first term on the right hand side of the inequality is exactly the entropy, i.e., the minimum encoding length according to Shannon. In fact, if we divide the above equation by n to get the per-symbol coding length, we see that the inequality asymptotically becomes Shannon's inequality converted to per-symbol coding length. In Shannon's theory the optimal code length for x_i is $-\log_2(p_i)$. Rissanen's theory is a generalization of this and the optimal code length for x^n asymptotically equals $-\log_2 f_\alpha(x^n)+(\frac{1}{2}-\varepsilon)k \log_2 n$, the non-predictive complexity. Thus, the non-predictive complexity is asymptotically equal to the optimal coding length. It can also be shown that the predictive complexity is asymptotically optimal.

As mentioned previously, no partial recursive lower bounds on descriptive complexity exist. Note that the lower bound mentioned above is not a lower bound in this sense. Rissanen's stochastic complexity is computable and so there is no need to compute lower bounds. This lower bound is a theoretical lower bound on the complexity with respect to all conceivable coding schemes based on the actual distribution of the data.

3.2. Implementation

3.2.1. MDL for Gaussian ARMA Models

Now we consider using MDL to select the order of ARMA models. ARMA models are typically used for prediction and so we employ the predictive complexity to select the model order. We consider a Gaussian excitation source, $\langle e_n \rangle$. Recall that in an ARMA model with parameters $a=(a_1,...,a_p)$ and $b=(b_1,...,b_q)$, the value at step t is defined as:

$$x_t = \sum_{i=1}^{p} a_i \, x_{t-i} + \sum_{i=1}^{q} b_i \, e_{t-i} + e_t$$

with the requirement that the filter is stable and has stable inverse. Now define $\hat{x}_t$:

$$\hat{x}_t = \sum_{i=1}^{p} a_i \, x_{t-i} + \sum_{i=1}^{q} b_i \, e_{t-i}$$

Then:

$$x_t = \sum_{i=1}^{p} a_i \, x_{t-i} + \sum_{i=1}^{q} b_i \, e_{t-i} + e_t = \hat{x}_t + e_t$$

or:

$$e_t = x_t - \hat{x}_t$$

each e_{t-i} can be determined from x^t and thus, so can $\hat{x}_t$. Substituting into the conditional density:

$$f_\alpha(x_{t+1} \mid x^t) = f_\alpha(\hat{x}_{t+1} + e_{t+1} \mid x^t)$$

Note that $\alpha=(p;a_1,...,a_p;q,b_1,...,b_q)$. Since $\hat{x}_{t+1}$ is determined by x^t and e_{t+1} is zero mean, Gaussian and independent of x^{t-1}, we have:

$$f_\alpha(x_{t+1} \mid x^t) = \frac{1}{\sqrt{2\,\pi}\,\sigma_e} \exp\left(\frac{-(x_{t+1} - \hat{x}_{t+1})^2}{2\,\sigma_e^2}\right)$$

In order to determine the predictive complexity, $\hat{x}_{t+1}$ is replaced by $\hat{x}_{t+1}(t)$, the estimate based on the model chosen from the data up to time t, $\hat{\alpha}(t)$:

$$\hat{x}_{t+1}(t) = \sum_{i=1}^{p(t)} a_i(t) \, x_{t+1-i} + \sum_{i=1}^{q(t)} b_i(t) \, e_{t+1-i}$$

The predictive complexity is therefore the sum of the squared errors of the estimates at each time t with some additive constant :

$$-\sum_{t=0}^{n-1} \log_2 f_{\hat{\alpha}(t)}(x_{t+1} \mid x^t) = \sum_{t=0}^{n-1} \left(\log_2 \left(\sqrt{2\,\pi}\,\sigma_e \right) + \frac{(x_{t+1} - \hat{x}_{t+1}(t))^2}{2\,\ln(2)\,\sigma_e^2} \right)$$

This is the exact value of the predictive complexity, but, in choosing a model based on this complexity, we can disregard the additive and multiplicative constants since they are constant with respect to the minimizations in question. The expression which is minimized to determine the number of parameters k (step 2 of Fig.3) is the sum of the actual squared errors and Rissanen calls it the accumulated prediction errors. Now fix p, q, a_i and b_i and define:

$$\hat{e}_t = \sum_{i=1}^{p} a_i\, x_{t-i} + \sum_{i=1}^{q} b_i\, \hat{e}_{t-i}$$

For fixed p and q, to determine $a_i(t)$ and $b_i(t)$ (step 1 of Fig. 3), the following is what must be minimized:

$$\sum_{t=0}^{n-1} \hat{e}_t^2$$

Note that this is exactly the ubiquitous least squares criterion. Thus, for Gaussian ARMA processes and fixed p and q, the MDL criterion corresponds with the least squares criterion. At time t, we choose the number of parameters p(t) and q(t) which minimizes the complexity with p and q held fixed over time, which is the sum of the squared errors based on the models $a_i(t)$ and $b_i(t)$ for that p and q, or the accumulated prediction errors.

3.2.2. Experimental Results for AR Models

In order to evaluate the MDL criterion, I experimentally tested it for Gaussian AR processes. The data was taken from AR and MA models and the fitted models were AR since the minimization is easier to compute (MA process estimation requires nonlinear programming). The AR process estimation was implemented using the least squares lattice algorithm[7] which works well in conjunction with MDL since models with varying order can be computed iteratively and since the a priori estimation errors are available (these are used to determine the predictive complexity and the accumulated prediction errors). Also, it is easy to test for stability for lattice filters since they are modular and stability can be tested for each module separately. Further implementation details can be found in [2].

In fitting an AR model to a sequence of data, it is important to restrict attention to only stable IIR filters. The theory behind MDL requires that the parameter space be compact. The space of coefficients of IIR filters is a compact space. In fact, there is no reason why MDL should not work on unstable models as long as a compact subset of Euclidean space is chosen for the parameter space (the set of all unstable models is all of Euclidean space which is not compact). However, in practice, MDL shows poor performance in choosing model order for stable data when the parameter space is large and includes unstable models. The convergence was very slow for such model classes.

Figures 4, 5, 6 and 8 show the results of the accumulated prediction errors for AR models of different orders and with data generated by a different model for each figure. In step 2 of the predictive MDL model choice, we choose the model order for which the accumulated prediction error is minimum. In all of these results, the variance of the white noise process is 1. This is the theoretical asymptotic minimum for the accumulated prediction errors. If the model parameters are estimated exactly for a sufficiently long period of time, then the accumulated prediction errors will fall below $1+\varepsilon$ for any $\varepsilon>0$. Figures 4 and 5 demonstrate the convergence of the accumulated prediction errors to the desired relative values (with the accumulated prediction errors of the correct model order being smallest). Figure 4 shows the accumulated prediction errors for models of orders 1 through 4 where the data is generated by an AR(2) model with coefficients $a_1=0.25$ and $a_2=0.5$. Figure 5 shows the accumulated prediction errors for models of orders 1 through 4 where the data is generated by an AR(3) model with coefficients $a_1=0$, $a_2=0.375$ and $a_3=0.5$. Note that these coefficients were chosen by setting each module of the lattice filter to be half way in between the stability boundary and 0. This is to prevent the coefficients from being too large and the model becoming unstable and also to prevent the coefficients from being too small and the model degenerating to a lower order model. In Figures 4 and 5, the accumulated prediction errors for the order 2 model becomes the smallest at roughly time step 45. In Figure 5, the accumulated prediction error for the order 3 model becomes smallest again at roughly time step 45. Thus, in these experiments, MDL chooses the correct model based on roughly 45 data points (n=45).

Figure 4

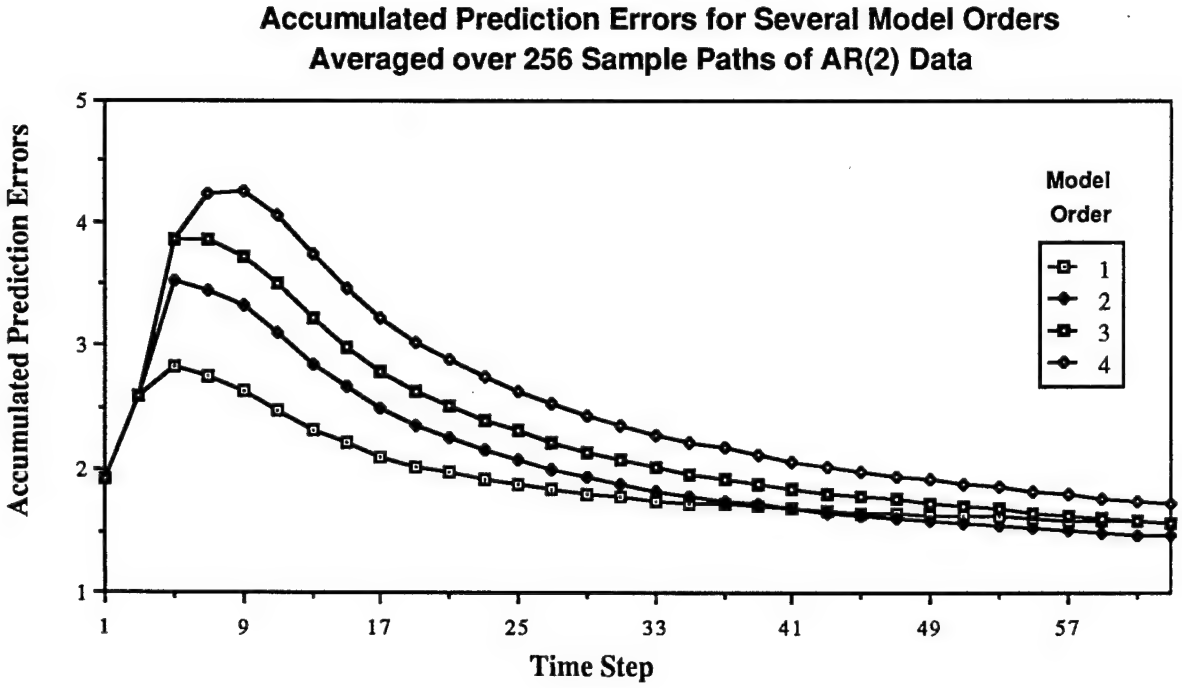


Figure 5

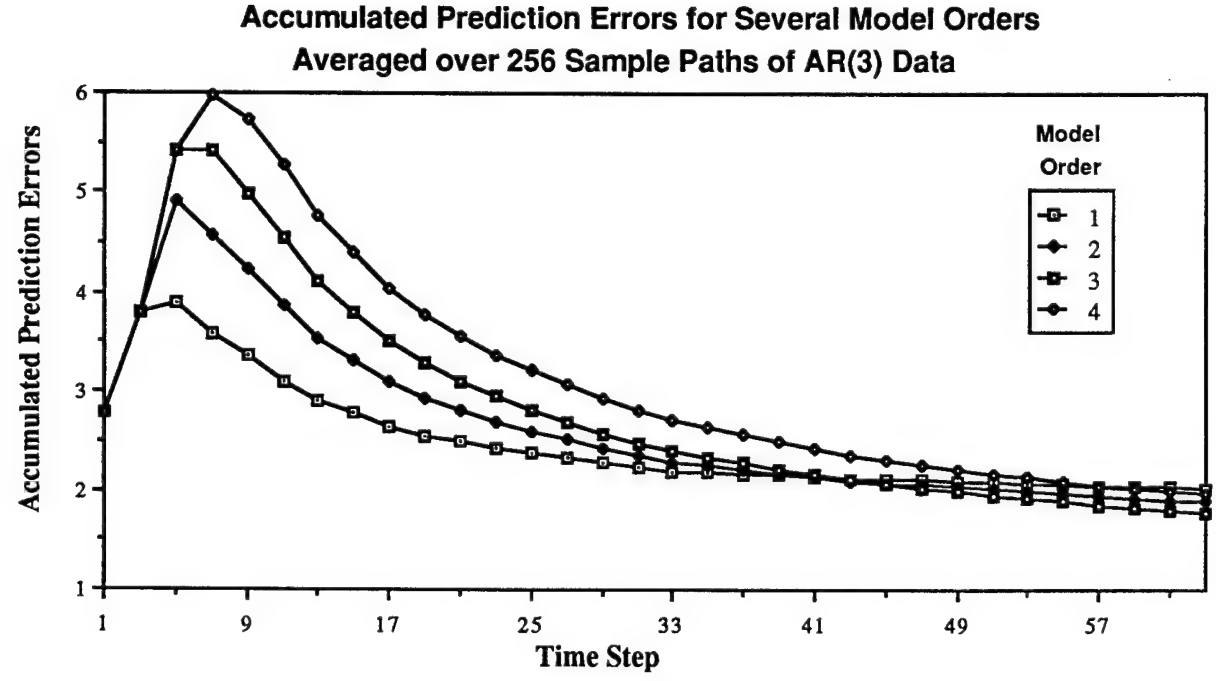


Figure 6 shows the accumulated prediction errors for models of orders 1 through 4 where the data is generated by an AR(2) model with coefficients a_1=0.4375 and a_2=0.125. Note that this model is more nearly degenerate than that of Figure 4. Since the second coefficient is relatively small, the model can be approximated fairly well by an AR(1) model. In other words, given the same input sequence, the filters of the AR(2) model and of the AR(1) model to which the AR(2) model degenerates would yield similar output sequences. This is demonstrated in Figure 7. Figure 7 shows a sample path from the AR(1) model to which the AR(2) model nearly degenerates as well as a sample path from the AR(2) model. Both sample paths are generated by using the same white noise sequence. Thus, the sample paths have the same likelihood in their respective models. The figure shows that even out to 1000 time steps, it is difficult to distinguish between the AR(1) and AR(2) models. Thus, it requires roughly 700 steps for the accumulated prediction error of the AR(2) model to become the minimum. Before this point, the accumulated prediction errors are smallest for the AR(1) model.

Figure 6

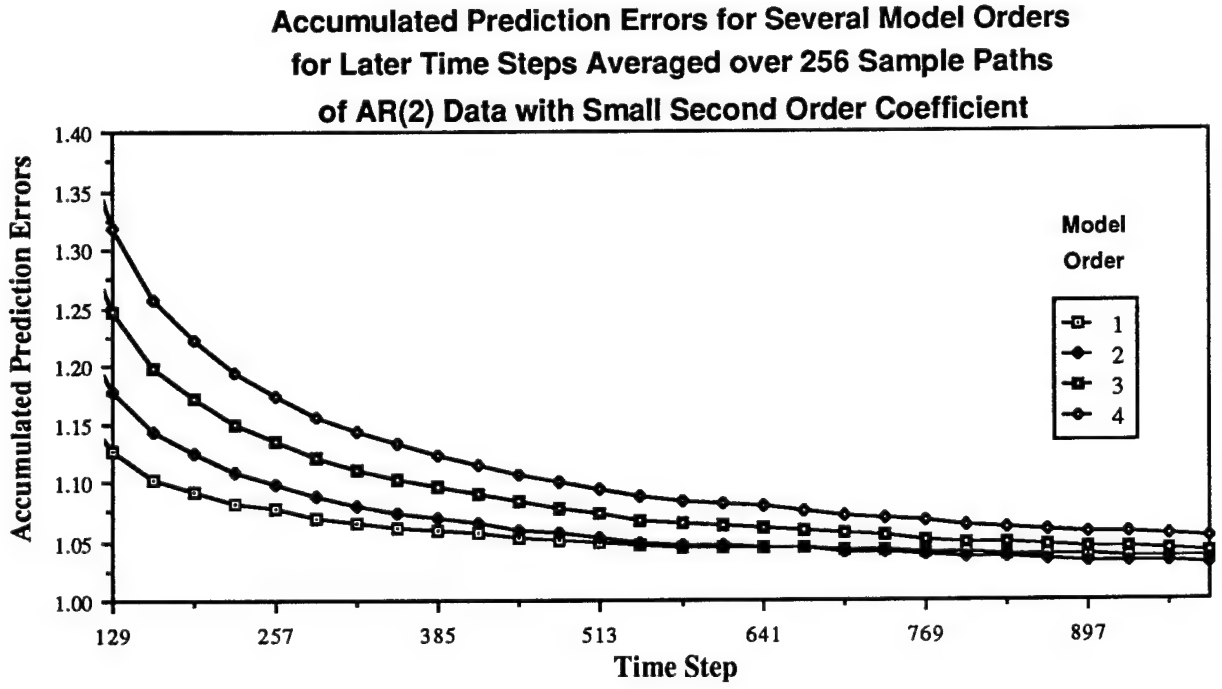


Figure 7

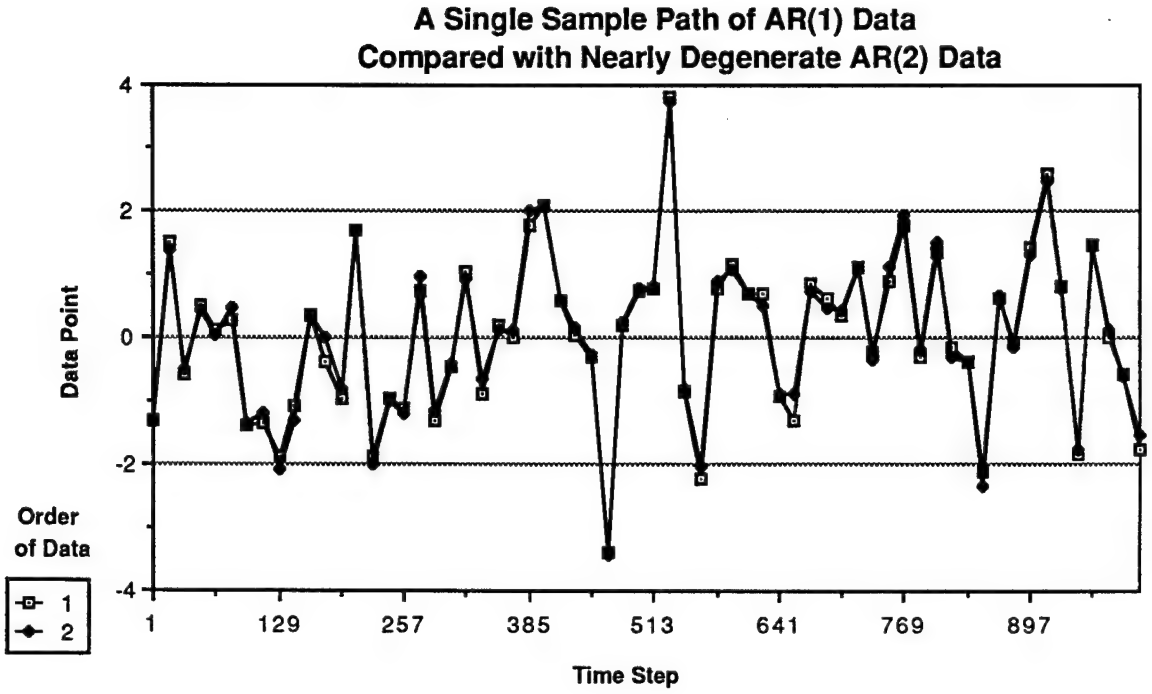
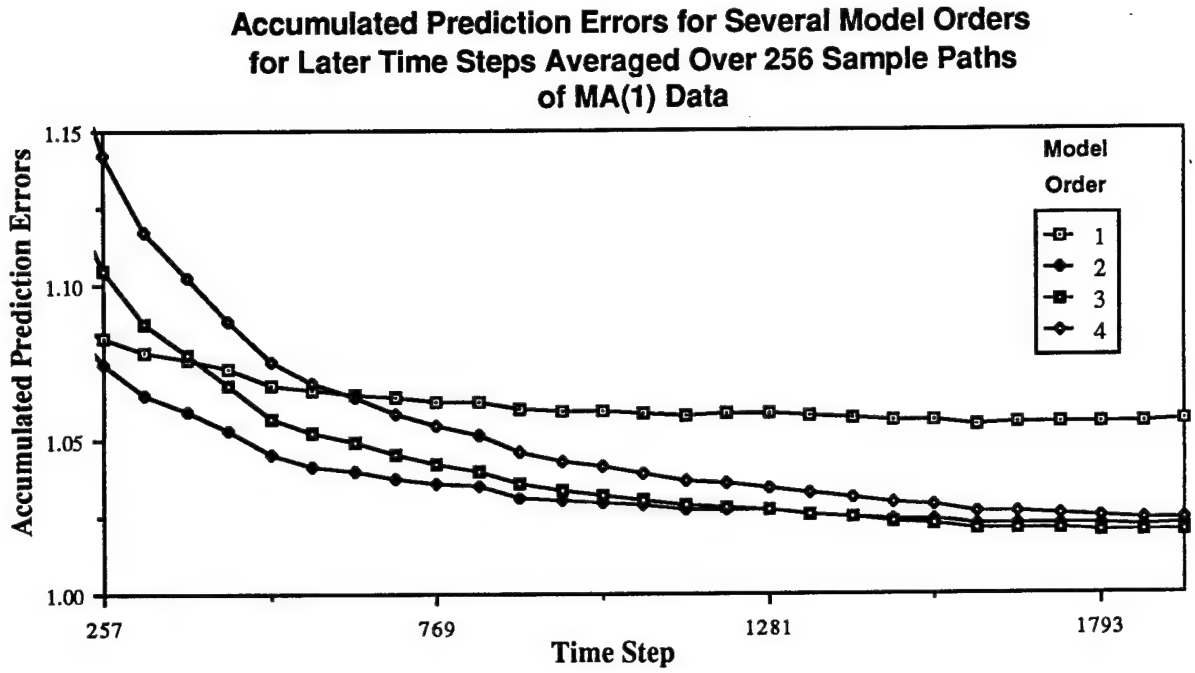


Figure 8 shows the accumulated prediction errors for models of orders 1 through 4 where the data is generated by an MA(1) model with coefficient b_1=0.5. In this case, none of the models in the chosen model class of AR models is the true model which generates the data. In fact, any MA model is the limit of a sequence of AR models with increasing order (this may be informally viewed as an infinite order AR model except AR models are only defined for finite orders). Thus, the AR model with the highest possible number of parameters should be chosen asymptotically. Figure 8 shows the accumulated prediction errors for the AR(3) model becoming smaller than those for the AR(2) model. Also, just before the first time step shown in the figure, the accumulated prediction errors for the AR(2) model overtook those for the AR(1) model. Finally, as can be extrapolated from the figure, the accumulated prediction errors for the AR(4) model overtake those for the AR(3) model at roughly time step 6000.

Figure 8



3.3. Discussion

The MDL principle is essentially an application of the philosophy behind descriptive complexity theory to statistical inference. It overcomes the problems of computability which occur in the more general descriptive complexity and, in this sense, it is a more practical theory. However, there may be cases when one should not perform statistical inference by minimal encoding length. In the case of Shannon's entropy, the applications are clear, one uses minimal encoding for transmission of information in order to minimize the use of the channel (the communication medium) which is presumably the limiting resource. In certain cases, we may have an explicit loss function which may differ from the channel use loss function which is used implicitly in Shannon's theory. In such cases, the encoding length is not the criterion which should be minimized (since we have an explicit loss function). However, when the loss function is not known, the minimum description length approach still maintains the highly intuitive theoretical foundation of performing inductive inference by minimal encoding. Within this foundation, it provides a consistent framework in which to handle statistical inference with only a small number of assumptions.

The choice of the class of models in MDL is roughly equivalent to the choice of interpreter in descriptive complexity theory. The difference is that in the case of MDL, we choose statistical models of which there is a wide variety which have been well developed in the literature. Rather than arbitrarily choosing an interpreter, we can choose from among many standard statistical models. These commonly used statistical models are usually those for which analysis or computation is particularly feasible. However,

there are several aspects of model choice which cause MDL to be limited in scope. First, Rissanen limits his theory to sequences of real numbers. Of course, something as general the minimum description length can be defined for data other than sequences but Rissanen only develops the specific theory for such sequences. For example, in computer vision, where the data is indexed by two variables and there is no inherent order in the index set, it is unnatural to describe the model as a sequence of reals. Secondly, as Dr. Mintz points out, because of the choice of a parametric model class, it is not clear how to go about handling the data in a non-parametric or robust manner. Perhaps one of the greatest strengths with the MDL principle is that, given two model classes, we can determine which class better fits (describes) the data in order to choose between them. However, MDL gives us no guidance as to how to choose the original model class from the set of all model classes.

Another potential source of controversy with MDL is the discretization of the models. The the data space of the models is discretized with fixed precision r. This is equivalent to assuming that the data is represented as fixed point numbers since the precision of floating point numbers is dependent upon the size of the numbers. Thus, if the data is actually in the form of floating point numbers (for example, data from a digital voltmeter), certain modifications would be required. This should not occur in most cases though, because most data is measured on a single scale. Another problem with discretization of the model is that the discretized model could be significantly different from the original continuous model. For example, the discretized model could have completely different extrema. This is a problem because one would like to optimize the continuous model so that calculus can be used rather than extensive searches. However, if the discretized version of the model does in fact differ significantly from the continuous version, then it suggests that the model was chosen with detail on a scale finer than the precision of the data. Thus, perhaps this problem suggests poor choice of model class rather than a problem with discretization of the model.

As with descriptive complexity theory, there may be other important factors which are not accounted for in MDL. For example, one model may provide an extremely good encoding of the data but prediction based it might be computationally infeasible. The advantage of MDL over descriptive complexity theory is in providing a computable inference procedure and yet it does not consider computational feasibility in its comparison of models. It is essentially up to the practitioner to weigh computational feasibility versus description length of models. This is natural since the computational feasibility is an aspect of the algorithms used rather than the models chosen (for any one model,there may be many algorithms which can be used). Again, it is contrary to a general theory of inductive inference to include details about the specific situation at hand. However, this may be a consideration in many practical instances.

The previous section demonstrates the application of the predictive MDL principle to AR processes. In all cases, the procedure gave the appearance of converging asymptotically to the correct choice of models. In the simplest cases, the convergence was rapid with the procedure choosing the correct model order in roughly 45 time steps. For the more tricky case of a nearly degenerate model, the convergence was slower requiring on the order of 700 time steps to choose the correct model order. For the early time

steps, the procedure choose the lower order model to which the higher order model degenerates. This may actually be more appropriate since, as demonstrated by the graph of the sample paths, the lower order model could be used to perform accurate predictions of the data. As the procedure obtained more data, it gradually determined the correct model order. Thus, although convergence was slow for the nearly degenerate case, it is not clear that faster convergence could be achieved nor that it is desirable in this case. The last case discussed was when the model class is inappropriate in that the data are not generated from a model in the class. In this case, the procedure choose appropriate low order models for the early time steps and gradually increased the model order to find closer approximations to the true model. One final point is that , although the results are not given here, the MDL criterion seems to have difficulty in determining the order of stable AR models when unstable models are included in the model class. Of course, such unstable models are rarely used at present and so there is little reason to include them in the model class. But one should keep in mind that MDL does have limitations.

MDL is one of the few principles of statistical inference which can help in determination of the number of parameters (for ARMA processes, for instance). Early in this report, it was mentioned that maximum likelihood and the least squares criterion will always choose, if possible, the distribution which gives all the mass to the data. Let us consider this distribution with the MDL criterion. Since the distribution gives probability one to the data, there is no cost in describing the data within the model, however, we must also describe the model. The model must be parameterized over each possible data sequence (since it gives probability one to the actual data sequence and this sequence is unknown beforehand unless there is significant prior knowledge) and so the data itself must be encoded to describe the model. Thus, this model has no value in the MDL sense, that is, it does not compress the data at all.

Another principle which can be used to determine the number of parameters is Akaike's information criterion A (AIC)[1], which is both a pre-cursor and an alternative to MDL. Akaike's AIC is based on the idea that if the estimating parameter is close to the actual parameter, then the log likelihood forms an estimate of the closeness of the model to the true model. It turns out to be similar to the Rissanen's non-predictive complexity. The theoretical justification of Akaike's criterion is perhaps not as strong as that for MDL. Further, Rissanen proves that MDL is provides consistent estimates, that is, estimates which converge to to the true model asymptotically, and that AIC does not provide consistent estimates. On the other hand, an empirical study[7] suggests that AIC may yield better results for small numbers of data points than non-predictive MDL for ARMA processes.

4. Minimum Description Length for Image Segmentation

4.1. Introduction

LeClerc[3] applies the minimum description length principle to the problem of image segmentation. Thus, we choose a set of models of segmented images and for any given image, select from among these models, that which allows the original image to be described as concisely as possible (including the description of the model). Of course, as mentioned, much of Rissanen's MDL is restricted to sequences of data but LeClerc's approach relates to MDL in its general philosophy. He suggests that problems in computer vision in general can be handled with the MDL philosophy. The minimum description length provides an objective criterion on which to base computer vision algorithms. LeClerc argues that this is after all what we are attempting to do in computer vision; find simple models of images.

LeClerc demonstrates an interesting equivalence between MDL and maximum a posteriori probability estimation (actually, these ideas are implicit in some of Rissanen's work but LeClerc explicitly discusses them). In MDL, we attempt to describe the data by describing the model and then describing the data within the model:

$$L(x^n) = L(\theta) + L(x^n \mid \theta)$$

According to Shannon, we know that if the model and the data given the model are discrete random variables, then the above will be equivalent to:

$$L(x^n) = -\log_2 g(\theta) - \log_2 f(x^n \mid \theta) = -\log_2 \left(g(\theta)\, f(x^n \mid \theta)\right)$$

where $f(x^n|\theta)$ is the distribution of the model with parameter θ, $g(\theta)$ is the prior distribution for the parameters and the model and the data are encoded optimally. We choose the model, θ, which minimizes this total description length. But by taking the exponential of the above, we see that this minimization is equivalent to the following maximization:

$$g(\theta)\, f(x^n \mid \theta)$$

which is equivalent the maximum a posteriori method of estimation. Note, however, that the difference between Rissanen's formalism and maximum a posteriori as it is used in practice, is that Rissanen also codes for the number of parameters which is the equivalent of giving a prior on the number of parameters.

4.2. The Model Classes

4.2.1. The Piecewise-Constant Model Class

In order to apply MDL, LeClerc defines a model class for segmented images. The model class is gradually expanded to include models of increasing generality. The simplest model class used is the class of piecewise-constant intensity images. Rather than giving a prior distribution on the class of models, LeClerc equivalently defines a method of encoding the models. LeClerc suggests describing a piecewise-constant image by describing the boundary of each segment with a chain code (a list of directions) and describing the constant intensity value within the region. LeClerc then assumes that the data given the model is distributed normally with known variance.

In the case of vision, the data has two spatial dimensions. We will denote data by $x^{nxn}=(x_{1,1},...,x_{1,n},...,x_{n,1},...,x_{n,n})$ where $x_{i,j}$ is the value of the intensity at spatial location (i,j). Let the model be denoted by $u^{nxn}=(u_{1,1},...,u_{1,n},...,u_{n,1},...,u_{n,n})$ where $u_{i,j}$ is the value which of the model at spatial location (i,j). The model is piecewise-constant and so the boundary of the segments occurs between any two points which have different values. The outer boundary need not be described since this is always included in the boundary of the corresponding region. The length of the boundary, therefore equals the number of points and neighbors which have different values divided by 2 (since, assuming 4 point neighborhoods, each boundary point will be counted twice, once from each side):

$$\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)\,\varepsilon\,N(i,j)}\left(1 - \delta(u_{i,j} - u_{k,l})\right)$$

where N(i,j) is the set of neighbors of (i,j) and $\delta(x)$ is the Kronecker delta function, which equals 1 when x=0 and 0 elsewhere. In order to encode a region, we must give the the starting point of the chain code of the boundary, the links of the chain code and also the constant value within the region. The encoding length of the links of the chain code for the boundary equals its length times the average length of encoding each link in the chain, which we will denote by b (b will be roughly $\log_2 3$ for 4 pixel neighborhoods). Thus, the cost of encoding the chains of all segments is just b times the total length of the boundary which is given above. LeClerc assumes that the encoding costs of the starting point of the chain and the constant value within the region are averaged into the global constant, b, by which the above total boundary length is multiplied. This is an approximation but the actual value is difficult to calculate locally since it depends on the number of regions and also, it is relatively small compared to some of the other terms in the description length. Thus, the length of encoding the model is taken to be the total length of the boundary, given above, times a constant b.

Now we must consider the length of encoding the data given the model. Assuming the data given the model is independent and normally distributed around the model value with known, constant variance σ, the optimal encoding length of the data given the model will be:

$$-\log_2 \prod_{(i,j)} \left(\frac{1}{\sqrt{2\,\pi}\,\sigma} \exp\left(-\frac{(x_{i,j} - u_{i,j})^2}{2\,\sigma^2} \right) \right) = \sum_{(i,j)} \left(\log_2\left(\sqrt{2\,\pi}\,\sigma\right) + \frac{(x_{i,j} - u_{i,j})^2}{2\,\ln(2)\,\sigma^2} \right)$$

Note that this noise is meant to characterize features such as texture and camera noise which are not otherwise included in the models. We now find the model to minimize the total encoding length which will be, ignoring constants:

$$\frac{b}{2} \sum_{(i,j)} \sum_{(k,l)\,\varepsilon\,N(i,j)} \left(1 - \delta(u_{i,j} - u_{k,l})\right) + \sum_{(i,j)} \frac{(x_{i,j} - u_{i,j})^2}{2\,\ln(2)\,\sigma^2}$$

Because of the δ functions in the above, it will be difficult to minimize with common optimization procedures such as gradient descent or simulated annealing. Therefore, LeClerc implements a special optimization procedure to minimize the above which will be discussed in the next section. Note that this objective function has similarities with those used in regularization[11].

4.2.2. The Piecewise Smooth Model Class

LeClerc then extends the model class to piecewise smooth images. In this case, piecewise smooth means that the function can be approximated by a low-order polynomial. This assumption is reasonable since, a class of functions which is particularly smooth is the analytic functions which can be approximated with the low order terms of the Taylor series in most cases. LeClerc chooses to describe the polynomials locally. Thus, the model at point (i,j) is a two dimensional polynomial centered at (i,j) which has the following form:

$$u_{i,j}(x,y) = \sum_{m=0}^{M} \sum_{n=0}^{M-m} u_{i,j}^{m,n} (x - i)^m (y - j)^n$$

where M is the maximum degree of polynomials (that is, maximum sum of degrees in each term) being used and $u_{i,j}^{m,n}$ is the coefficient of the term which is degree m in x and degree n in y (actually, LeClerc includes a factor of $\frac{1}{m!\,n!}$ as in the Taylor series but this is unimportant since it can be included in the

coefficient $u_{i,j}^{m,n}$). LeClerc uses polynomials which have a maximum total degree rather than a maximum degree in each variable so that the class of models is rotationally invariant.

Now we consider coding the model. As before, we must describe the boundary across which the model changes. The model is the same at two points if and only if all the derivatives of the polynomials are the same at the midpoint (the polynomials are not the same but are translated to be centered at their corresponding spatial locations and so they are equal when translated to the midpoint). The derivatives can easily be obtained analytically in terms of the coefficients of the polynomials. Now the total region length will be the half the number of points and neighbors which differ in any of their derivatives at the midpoint:

$$\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)\,\varepsilon\,N(i,j)}\left(1-\prod_{p=0}^{M}\prod_{q=0}^{M-p}\delta\left(u_{i,j}^{p,q}\left(\frac{i+k}{2},\frac{j+l}{2}\right)-u_{k,l}^{p,q}\left(\frac{i+k}{2},\frac{j+l}{2}\right)\right)\right)$$

where $u_{i,j}^{p,q}(x,y)$ is the mixed p-th partial derivative with respect to x and q-th partial derivative with respect to y of $u_{i,j}(x,y)$:

$$u_{i,j}^{p,q}(x,y)=\frac{\partial u_{i,j}(x,y)}{\partial x^p \partial y^q}=\sum_{m=0}^{M}\sum_{n=0}^{M-m}u_{i,j}^{m,n}\,\frac{m!}{p!}(x-i)^{m-p}\,\frac{n!}{q!}(y-j)^{n-q}$$

As in the previous case, we multiply the total boundary length by a factor b which represents the average length of encoding a link in the chain code. However, in this case, we do not include the length of encoding the parameters (polynomial coefficients) of the region into the constant b because unlike the previous case, the length of encoding the coefficients may be a significant factor and may vary between regions. In this case, we encode the coefficients of the polynomials separately and allow for a different degree polynomial in each region. We encode all coefficients for terms with a given sum of degrees if any of them are non-zero. Thus, the total number of coefficients which must be encoded for a single region is:

$$\sum_{m'=0}^{M}(m'+1)\left(1-\prod_{m=0}^{m'}\prod_{n=0}^{m'-m}\delta(u_{i,j}^{m,n})\right)$$

where (i,j) is any point in the region. Here m' is the index for the sum of degrees and we multiply by m'+1 since there are this many coefficients with sum of degrees m'. In order to get the encoding length

of the parameters for the region, we multiply the above by a constant d, the length of encoding a single non-zero coefficient. This is the encoding length for a single region. Once again, it is difficult to locally calculate the total encoding length for all the regions since this is dependant upon the number of regions. Thus, we approximate it by calculating the above at each point and multiplying it by a factor which should be roughly the reciprocal of the average region size, which we incorporate into the constant d. To get the total encoding length of the data, we add the length of encoding the region boundaries derived at the beginning of this section, with the length of encoding the coefficients derived above and the length of encoding the squared errors, derived at the end of the previous section. We again choose the model which minimizes the total encoding length.

4.2.3 Further Extensions

LeClerc further extends the class of models in two ways. First, the model of the noise is changed so that the variance is unknown and varies between regions (thus, the variance becomes piecewise constant, rather than constant as in the previous cases). The following modifications must be made to the coding length which is to be minimized. First, since the variance is now spatially varying, σ becomes $\sigma_{i,j}$ at point (i,j) everywhere in the equation for the coding length. Also, the constant which was ignored in the analysis of the squared errors must be included in this case since it is dependent upon the variance which is now unknown. Finally, changes in variance must be included as boundary points when calculating the region length. Again, LeClerc includes the small length required to encode the variance for each region into the constant in front of the region boundary length term.

The other extension which LeClerc makes to the model is to include a known point spread function for the image sensor. This involves a convolution with a known kernel of the estimates formed with the model (before the noise is added).

4.3. The Optimization Procedure

LeClerc introduces an optimization procedure to minimize the functionals involving δ functions which arise as the description length of partitioned images. The optimization method is one of a general class of numerical methods known as continuation or homotopy methods[17]. In a continuation method, one embeds the problem at hand into a class of problems indexed by some parameter, s, such that for one value of the parameter the problem is easily solved and for another value of the parameter, the problem is equivalent to the original problem. Also, it must be required that the solution varies continuously with s. In our case, the original problem is a minimization of a functional which will be denoted by $L(u^{nxn})$. We embed this functional into a class of functionals $L(u^{nxn},s)$ such that:

$$L(u^{nxn}, 0) = L(u^{nxn})$$

We require that $u^*(s)$, the minimum of $L(u^{nxn},s)$ at s, is continuous in s. Further, we choose $L(u^{nxn},s)$ so that it is easily minimized for large s (it has a unique global minimum).

Now we consider the optimization problem introduced in the previous sections. The main difficulty is with optimizing the δ functions. Thus, the functionals $L(u^{nxn},s)$ are formed by replacing the δ functions with Gaussian functions:

$$\delta(\Delta) \rightarrow \exp\left(\frac{-\Delta^2}{\sigma^2 s^2}\right)$$

where Δ is the difference in the models at two points (the difference in the constant value for the piecewise constant case and the difference in the derivatives of the polynomials at the midpoints for the piecewise smooth case). When s=0, this is equivalent to the original problem and as s gets large, the sharp valleys flatten out and the problem becomes easier to optimize. Thus, we start with a large value of s and multiply s by some fraction to gradually decrease it. For each value of s, we minimize the functional using the optimum from the previous value of s as the starting point. LeClerc optimizes the functional for each value of s by linearizing the derivative (which can be obtained analytically) of the functional and solving for a zero by using Gauss-Seidel iterations. This is the optimization procedure that LeClerc uses to minimize the functionals.

For the case when the variance is unknown and spatially varying, the above optimization procedure has difficulty in optimizing for the variance and the coefficients of the polynomials. Thus, LeClerc modifies the procedure so that it starts with an estimate of the variance and for each value s, finds the optimum value of u^{nxn} and then uses this to form the next estimate of the variance for each region.

There is another reason for the use of this choice of optimization procedure. The optimization procedure has an interpretation in terms of the discontinuities of the image. When the value of the Gaussian function shown above falls below a certain value, then a discontinuity is detected between the two points. When s is large, Δ must be large in order for a discontinuity to be detected. As s gets smaller, smaller variations in the models are detected as discontinuities. LeClerc defines the stability of a discontinuity as the first value of s at which it is detected. Discontinuities with high stability are those which are more easily detected. Thus, the stability of the discontinuity is a criterion for the scale at which the discontinuity can be detected. Also, it can be shown that, when s is large, the solution to the functional is roughly a linearly smoothed version of the image. As s becomes smaller, the smoothing becomes sharper and it does not spill over region boundaries. Thus, the procedure can be considered as a type of adaptive smoothing process. It starts out smoothing the entire image and gradually adapts (as s gets smaller) so that it doesn't smooth across detected discontinuities.

4.4. Discussion

The minimum description length approach to image segmentation follows from the previously discussed work on descriptive complexity and the minimum description length principle. Although it follows similar ideas to those of MDL, it is different in that the data are not sequences, i.e., single images have no inherent order or temporal index.

The work presented by LeClerc only represents one choice of model classes. For example, the noise is modeled as a normal distribution with unknown variance. It is not clear that this is an appropriate model class, particularly since, image data is typically bounded while the normal distribution is not. The problem is not serious since MDL is based on the idea of choosing an appropriate model from the model class rather than the exact model which may not exist. Nonetheless, a more appropriate model class such as a Beta distribution might yield better results (although the analysis might be less clean). However, in other cases, the model choices (chain code boundaries and polynomial regions) seem roughly to represent the intuitive ideas we have about segmentations. Furthermore, the class of models is defined so that the description length can be computed locally and so the procedure is well-suited for implementation on a parallel computer.

It is standard in the image segmentation field (as well as other areas of computer vision) to demonstrate a procedure by showing the resulting segmentations. The idea is that a human can generally tell what a good segmentation should look like. The results shown in LeClerc's paper[3] appear to be good compared to other techniques found in [6]. However, only a few results are actually presented. In fact, the two real images (as opposed to synthetic images) presented are coarsely sampled and generally lack details (there are no textures, the backgrounds are solid, etc.). Furthermore, one of the greatest potentials of minimum description length applied to computer vision lays in the ability to use the description length as an objective criterion with which to compare procedures in almost any area of computer vision. This criterion is precisely what is sought after in computer vision, a procedure performs well if and only if it produces a more concise description of the image. In fact, the ultimate goal in computer vision can be seen as finding a concise description of an image in some representation language, for example, we might want a description like "a tree with a wide trunk and red leaves" (although perhaps English is not the ideal representation language for images). Unfortunately, LeClerc does not take advantage of the minimum description length as an objective criterion. He does not give the resulting description lengths of the images which were processed which would give the reader a basis for comparison. Also, LeClerc requires that the representations of the data be complete in the sense that an image can be recreated exactly from its representation. The ultimate representation used by humans is not complete in the sense that humans typically cannot perfectly recreate images which they view. They can usually recreate only the "essential" information in the image from the model (in fact, the "essential" information may be context dependent and can vary between different people and would be difficult to formalize). Nonetheless, complete representations of images can in fact be a good model for early vision, in which all information is retained and the data is merely put in a more convenient form.

Many techniques for image segmentation are designed empirically. The approach presented here has the advantage that it is derived completely from a few simple principles. Another approach to image segmentation with a strong theoretical justification is the regularization approach. Regularization[12] is a general technique for solving ill-posed problems, that is, problems in which existence, uniqueness or stability of the solution is not guaranteed. The regularization method chooses an approximate solution which is stable with respect to changes of the initial data within some region. The regularization method requires choice of certain functionals (problem definition and regularizing functionals) which may be seen to be the analog of the choice of model class in the MDL approach. The problem is ultimately phrased as a problem in the calculus of variations. In certain cases, the regularization approach can be shown to be equivalent to maximum a posteriori estimation using Markov random field models[12]. One difference between the regularization approach to image segmentation and the approach presented here is that in regularization, the underlying image is chosen from the class of all piecewise smooth functions. It is not clear how this would be handled in MDL since it would be required to code functions from the class of all piecewise smooth functions. One would have to parameterize the functions and then discretize them as LeClerc did via Taylor series. However, it does not seem that this would be invariant to the choice of parameterization. Regularization has also been applied to several other areas of computer vision. The theoretical foundations of the MDL approach are somewhat stronger, in that, as mentioned previously, the descriptive complexity approach is an intuitive approach to inductive inference. Perhaps the ultimate test of these approaches will be their relative performance in real tasks.

Now we consider the optimization procedure used by LeClerc. If s is decreased to 0 sufficiently slowly, then the global minimum of the problem will be found. However, in practice, the rate of decrease and the stopping point are determined for computational feasibility and other reasons and so a global minimum is not guaranteed. In fact, stopping with s greater than 0 causes another problem. In the original formulation, the region boundaries occur where the models differ by any amount, which forces the regions to have closed boundaries. With the optimization procedure, the δ functions are approximated by Gaussian functions and so for any s greater than 0, the discontinuities are detected at points where the difference between the models of neighbors exceeds some threshold value. However, since s is not taken all the way to 0, the discontinuities found in an actual implementation do not have to form closed curves. The reason that LeClerc uses such a system is for computational feasibility and to ensure stability of the solution. By using this threshold, the solution is not sensitive to small perturbations in the data. Thus, LeClerc's approach is not purely MDL but is more of a hybrid between MDL, in which description length is the primary criterion, and regularization, in which solutions are chosen from the set of stable solutions. As $s \rightarrow 0$, the threshold loses importance since the models will be different only at discontinuities but the choice of a threshold is still somewhat arbitrary. Thus, the stability criterion has a degree of arbitrariness since it was included after the fact, rather than formulating the problem from the beginning so that it chooses a stable solution with the minimum description length.

5. Summary and Conclusion

We have introduced descriptive complexity approaches to inductive inference. The basic idea behind these approaches is that, in order to choose a model for a set of data, we choose the model which allows us to describe the data as concisely as possible. In the case of the very general descriptive complexity defined by Kolmogorov and others, this corresponds to choosing the shortest program which describes the data as a model for the data. This approach demonstrates an important correspondence between inductive inference and randomness. Random data is data for which cannot be induced, that is, no algorithm will allow prediction or equivalently compression of the data. Alternatively, random strings can be seen as those which possess no statistical regularities that can be found with an algorithm. However, this theory has the problem that the descriptive complexity cannot be computed by any algorithm. Also, the complexity is dependent on the programming language, when one considers the infinite number of possible programming languages.

In order to solve some of the problems associated with descriptive complexity, Rissanen developed the minimum description length principle. With MDL, a class of probabilistic models is chosen from which to choose a model for the data rather than all possible models as in descriptive complexity. From this class of models, the model which allows the most concise description of the data is chosen. This approach differs from other common approaches to statistical inference in that the description length of the model must be included in the description length since otherwise the data is not fully determined by its description. This allows classes of models in which the number of parameters can vary to be handled within the MDL framework. Experiment has demonstrated that the MDL predictive complexity works reasonably well for the important class of AR processes.

LeClerc applies the MDL principle to the problem of image segmentation. The class of models used by LeClerc consists of chain codes for region boundaries, polynomials for the region intensities and a normal distribution with unknown and spatially varying noise for the intensity noise. A special procedure is designed for the minimization of the description length because of the difficulty of the minimization. The results appear to provide good segmentations but LeClerc ignores the description length as an objective criterion of the appropriateness of the segmentation. Also, a result of making the optimization procedure computationally feasible is that the discontinuities do not form closed curves.

The descriptive complexity approaches to inductive inference define an intuitively pleasing approach to model identification. However, as this report demonstrates, the framework is quite general and care must be used in its application. Perhaps of all that can be said about descriptive complexity, this is the most important: it provides general guidelines to approaching inductive inference problems but the final outcome is ultimately dependent on how it is used in practice.

Primary References

1 Cover, Gacs and Gray (1989). "Kolmogorov's Contribution to Information Theory and Algorithmic Complexity". *The Annals of Probability* **17** (3) 840-865.

2 Rissanen (1986). "Stochastic Complexity and Modeling". *The Annals of Statistics* **14** (3) 1080-1100.

3 LeClerc (1989). "Constructing Simple Stable Descriptions for Image Partitioning". *International Journal of Computer Vision* **3** 73-102.

Bibliography

[1] Akaike (1974). "A New Look at Statistical Model Identification". *IEEE Transactions on Automatic Control* **19** (6) 716-723.

[2] Atteson (1990). "The Least Squares Lattice Algorithm and the Minimum Description Length Principle with Applications to Speech Modeling". Unpublished report.

[3] Box and Jenkins (1976). *Time Series Analysis: Forecasting and Control.* Holden-Day, Oakland, California.

[4] Chaitin (1966). "On the Length of Programs for Computing Binary Sequences". *Journal of the Association of Computing Machinery* **13** 547-569.

[5] Church (1940). "On the Concept of a Random Sequence". *Bulletin of the American Mathematical Society* **46** 254-260.

[6] Haralick and Shapiro (1985). "Image Segmentation Techniques". *Computer Vision, Graphics, and Image Processing* **29** 100-132.

[7] Haykin (1986). *Adaptive Filter Theory.* Prentice-Hall, Englewood Cliffs, New Jersey.

[8] Kolmogorov (1956). *The Foundations of Probability Theory.* Chelsea, New York.

[9] Kolmogorov (1965). "Three Approaches to the Quantitative Definition of Information". *Problems in Information Transmission* **1** 1-7.

[10] Martin-Löf (1966). "The Definition of Random Sequences". *Information and Control* **9** 602-619.

[11] Mumford and Shah (1985). "Boundary Detection by Minimizing Functionals, I". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 22-26.

[12] Poggio, Torre and Koch (1985). "Computational Vision and Regularization Theory". *Nature* **317** 314-319.

[13] Rissanen (1986). "Order Estimation by Accumulated Prediction Errors". *Essays in Time Series Analysis* 55-61. Applied Probability Trust, Sheffield, England.

[14] Rissanen (1987). "Stochastic Complexity". *Journal of the Royal Statistical Society* **49** (3) 223-239, discussion 253-265.

[15] Shannon (1948). "A Mathematical Theory of Communication". *Bell Systems Technical Journal* **27** 379-423.

[16] Solomonoff (1964). "A Formal Theory of Inductive Inference I/II". *Information and Control* **7** 1-22, 224-254.

[17] Stoer and Bulirsch (1980). *Introduction to Numerical Analysis.* Springer-Verlag, New York.

[18] Von Mises (1964). *Mathematical Theory of Probability and Statistics.* Academic, New York.

[19] Zvonkin and Levin (1970). "The Complexity of Finite Objects and the Development of the Concepts of Information and Randomness by Means of the Theory of Algorithms". *Russian Mathematical Surveys* **25** (6) 83-124.